

EXHIBIT 14

EXHIBIT G

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

Chart Detailing Defendant's Infringement of U.S. Patent No. 9,298,864

Wapp Tech Ltd. & Wapp Tech Corp. v. JP Morgan Chase Bank, N.A.

Case No. 4:23-cv-01137 (E.D. Tex.)

The Accused Instrumentalities include Xcode Developer Tools from Apple Inc. that Defendant uses to test applications for iOS mobile devices. Xcode Developer Tools include tools such as the Xcode integrated development environment (“IDE”), Network Link Conditioner, Instruments, and Simulator.

Based on the information presently available to them, Plaintiffs Wapp Tech Limited Partnership and Wapp Tech Corp. (“Wapp” or “Plaintiffs”) are informed and believe that Defendant directly and indirectly infringes U.S. Patent No. 9,298,864 (the “‘864 Patent”). Defendant directly infringes the ‘864 Patent when its employees, agents, and/or representatives use the Accused Instrumentalities to test applications for mobile devices. Upon information and belief, to the extent Defendant uses third parties in the testing process, it indirectly infringes the ‘864 Patent by actively inducing the direct infringement of third parties contracted to use the Accused Instrumentalities to test applications for mobile devices on Defendant’s behalf.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

TABLE OF CONTENTS

CLAIM 1.....	4
[1pre] A system for testing an application for a mobile device comprising:	4
1. Network Link Conditioner on iOS Device (e.g., iPhone 12).....	9
2. Network Link Conditioner on MacOS Computer/Xcode	12
[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;.....	18
1. Xcode Developer Tools Emulation/Simulation Overview	18
2. Xcode Developer Tool Display Overview	40
[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.	61
CLAIM 8.....	68
[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.....	68
CLAIM 13.....	78
[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.	78
CLAIM 14.....	82
[14] The system of claim 1, wherein the software is further configured to interact with a network to enable a user to update the software.....	82
CLAIM 17.....	85

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[17] The system of claim 1, wherein the software is further configured to enable a user to select from one or more network characteristics for testing how well mobile content performs on the mobile device..... 85

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

Claim 1

[1pre] A system for testing an application for a mobile device comprising:

The Accused Instrumentalities, such as Apple's Xcode Developer Tools (including at least Xcode IDE, Network Link Conditioner, Instruments, and Simulator), are a system for testing an application for a mobile device. Defendant tests its mobile banking applications through its use of the Accused Instrumentalities by, for example, executing compiled source code for the application and monitoring the execution of the compiled source code and creating and executing tests (including but not limited to through the use of unit tests).

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

Xcode

Build, **test**, and submit your app with Apple's integrated development environment.

Overview

Xcode is a suite of tools developers use to build apps for Apple platforms. Use Xcode to manage your entire development workflow — from creating your app to **testing**, optimizing, and submitting it to the App Store.

<https://developer.apple.com/documentation/Xcode/>, accessed on 2024/03/13 (annotated)

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

Xcode also includes several development tools to help you rapidly prototype and test your app. Use Simulator for rapid prototyping and testing your app in a simulated environment when a real device isn't available. Use Instruments to profile and analyze your app, improve performance, and investigate system resource usage. Construct 3D content with Reality Composer, train custom machine learning models with Create ML, and identify areas of your app that aren't accessible with Accessibility Inspector.

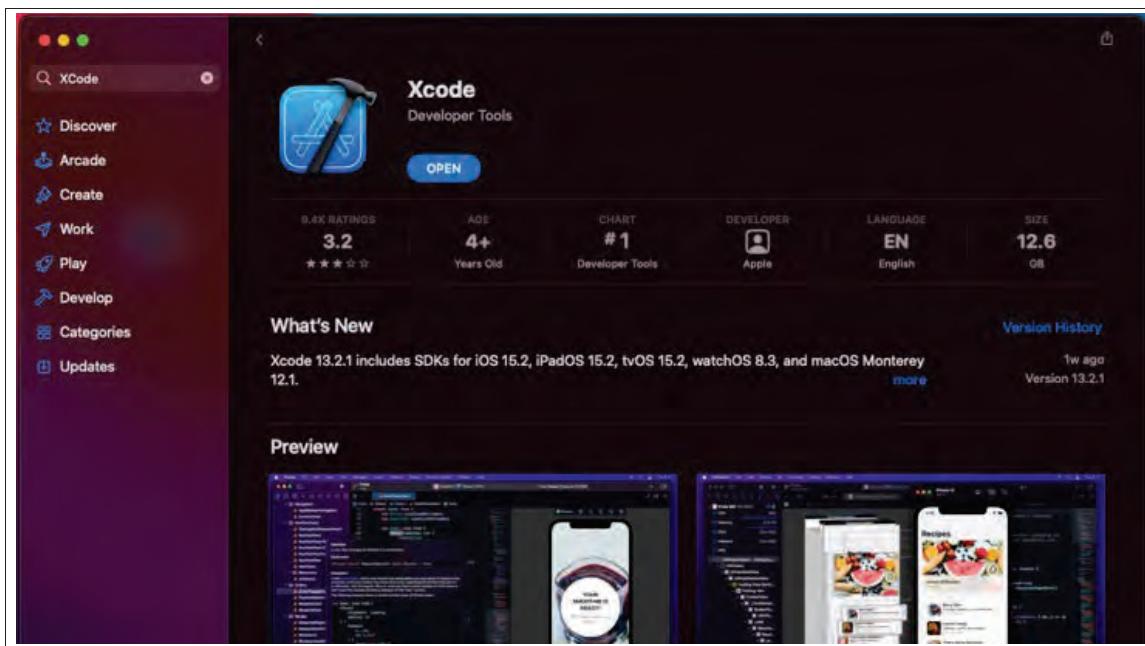
<https://developer.apple.com/documentation/Xcode/>, accessed on 2024/03/13 (annotated).

Xcode IDE is Apple's integrated development environment ("IDE")—it enables developers to test iOS applications by executing compiled source code for the application and monitoring the execution of the compiled source code. Xcode IDE itself runs on a computer running MacOS operating system. Applications tested through Xcode are either tested on a physical iOS device (such as an iPhone), or tested on the computer running Xcode using a simulator. Simulators for various iOS devices are included with Xcode, along with Instruments used to measure the performance of the application being tested.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

On information and belief, Defendant uses Apple's Xcode Developer Tools installed from a source such as Apple's App Store, shown below:

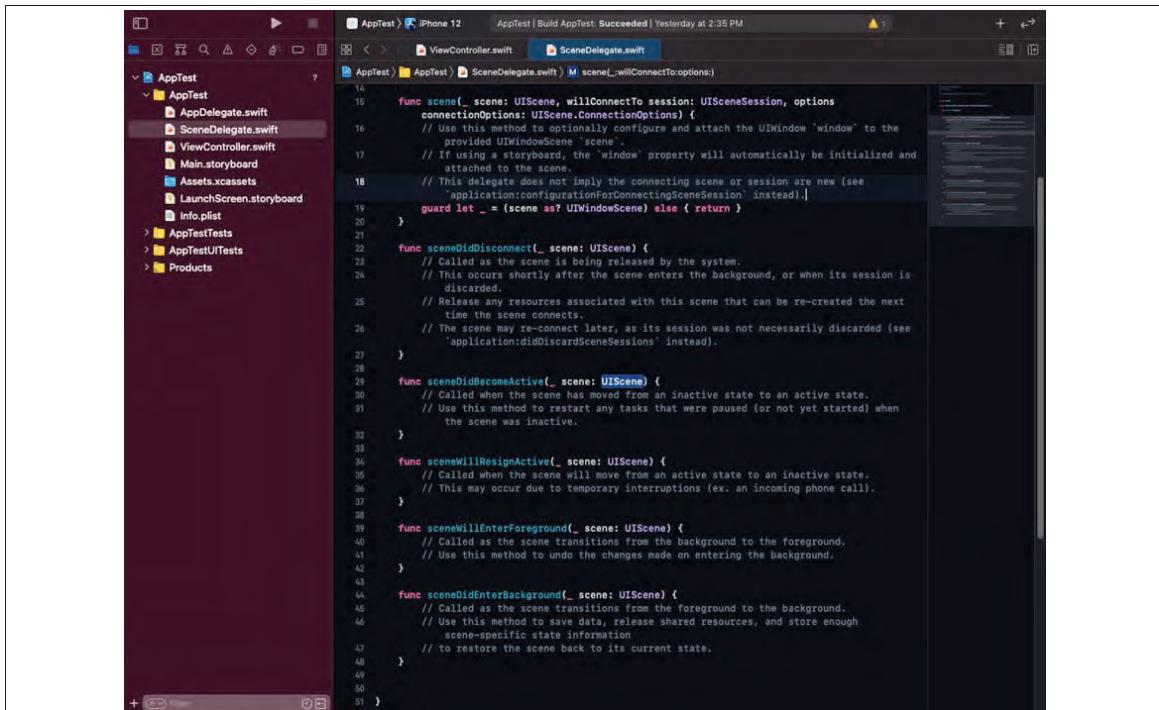


Installing Apple's Xcode Developer Tools results in the installation of infringing Accused Instrumentalities, including Xcode IDE, Instruments, and Simulator.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

An example of an Xcode IDE window from Apple's Xcode Developer Tools for reviewing source code files is shown below:



The screenshot shows the Xcode IDE interface. The left sidebar displays the project structure for "AppTest", including files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, AppTests, AppTestUITests, and Products. The right pane shows the source code for SceneDelegate.swift. The code defines several methods related to scene transitions:

```
14 func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
15     // Use this method to optionally configure and attach the UIWindow `window` to the
16     // provided UIWindowScene `scene`.
17     // If using a storyboard, the `window` property will automatically be initialized and
18     // attached to the scene.
19     // This delegate does not imply the connecting scene or session are new (see
20     // `application:configurationForConnectingSceneSession` instead).
21     guard let _ = (scene as? UIWindowScene) else { return }
22 }
23 func sceneDidDisconnect(_ scene: UIScene) {
24     // Called as the scene is being released by the system.
25     // This occurs shortly after the scene enters the background, or when its session is
26     // discarded.
27     // Release any resources associated with this scene that can be re-created the next
28     // time the scene connects.
29     // The scene may re-connect later, as its session was not necessarily discarded (see
30     // `application:didDiscardSceneSessions` instead).
31 }
32 func sceneDidBecomeActive(_ scene: UIScene) {
33     // Called when the scene has moved from an inactive state to an active state.
34     // Use this method to restart any tasks that were paused (or not yet started) when
35     // the scene was inactive.
36 }
37 func sceneWillResignActive(_ scene: UIScene) {
38     // Called when the scene will move from an active state to an inactive state,
39     // This may occur due to temporary interruptions (ex. an incoming phone call).
40 }
41 func sceneDidEnterForeground(_ scene: UIScene) {
42     // Called as the scene transitions from the background to the foreground.
43     // Use this method to undo the changes made on entering the background.
44 }
45 func sceneDidEnterBackground(_ scene: UIScene) {
46     // Called as the scene transitions from the foreground to the background.
47     // Use this method to save data, release shared resources, and store enough
48     // scene-specific state information
49     // to restore the scene back to its current state.
50 }
```

The above image shows an example window from Xcode IDE. The left pane displays different files which make up an application named “AppTest” (the application being tested). The right pane displays the source code for a file named “SceneDelegate.swift” (which contains source code related to certain visual elements of the sample application). This window is used to review source code, which is often part of software testing, where various tests are run as part of the app development process.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

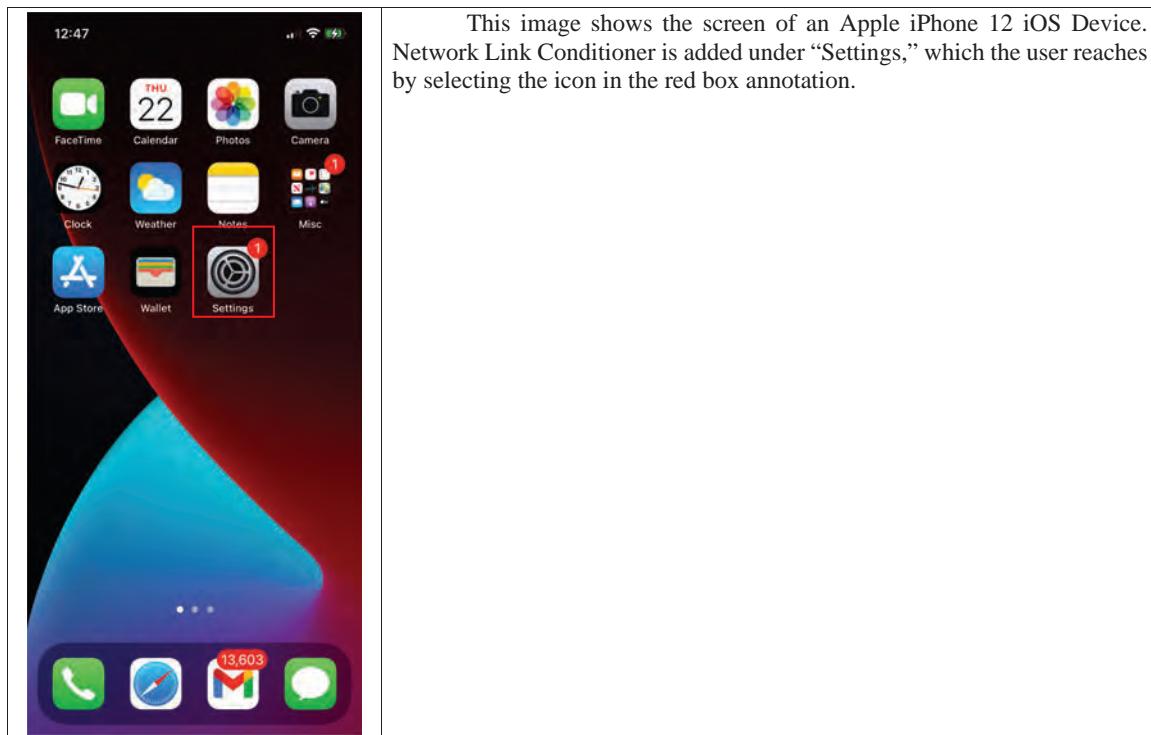
Network Link Conditioner is an application that simulates/emulates network characteristics such as bandwidth limitations, packet loss, latency, etc. Network Link Conditioner may be (1) installed/activated in the settings menu on an iOS device (such as an iPhone), for example, when the developer plugs in that iOS device to a computer running Xcode, or (2) installed on the computer running Xcode by selecting menu options from Xcode. Running Network Link Conditioner allows simulation/emulation of network characteristics.

1. Network Link Conditioner on iOS Device (e.g., iPhone 12)

The below screenshots demonstrate how a developer accesses Network Link Conditioner on an Apple iPhone 12 iOS Device. Note that the references to an iPhone 12 (and all other specific device models referenced throughout this document) are illustrative, and infringement is not limited to any specific model or version of iOS or MacOS device.

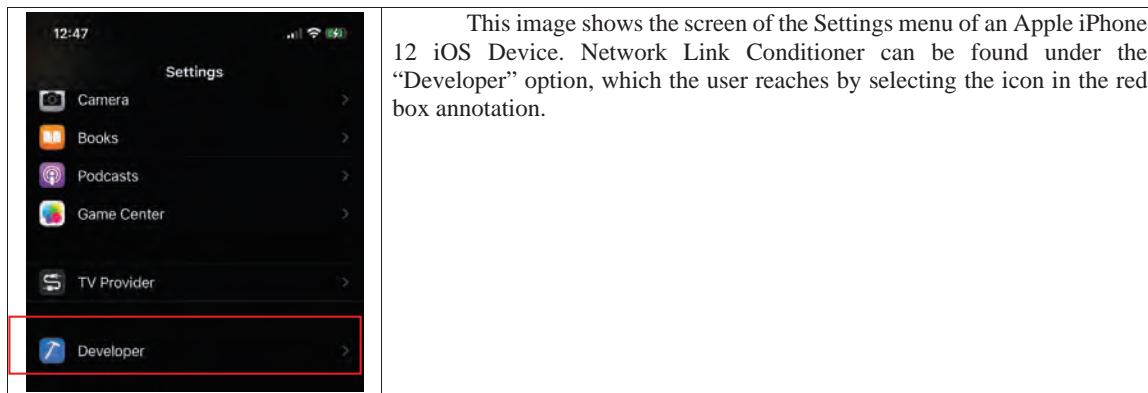
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:



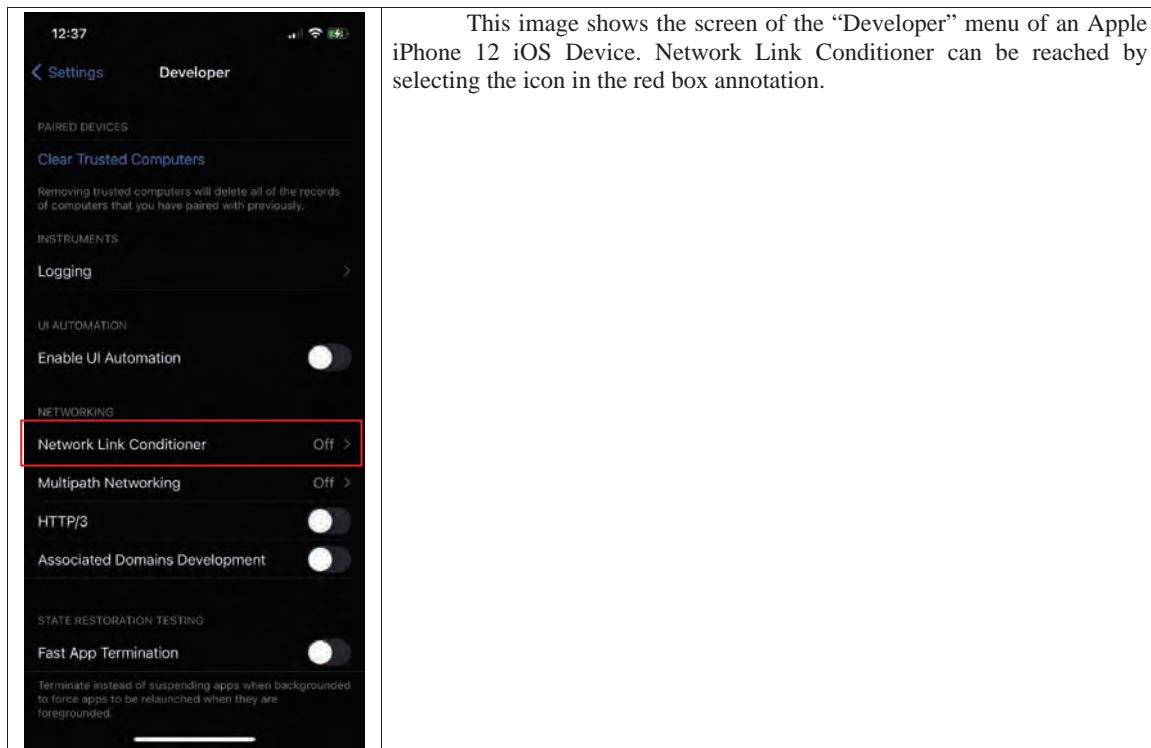
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

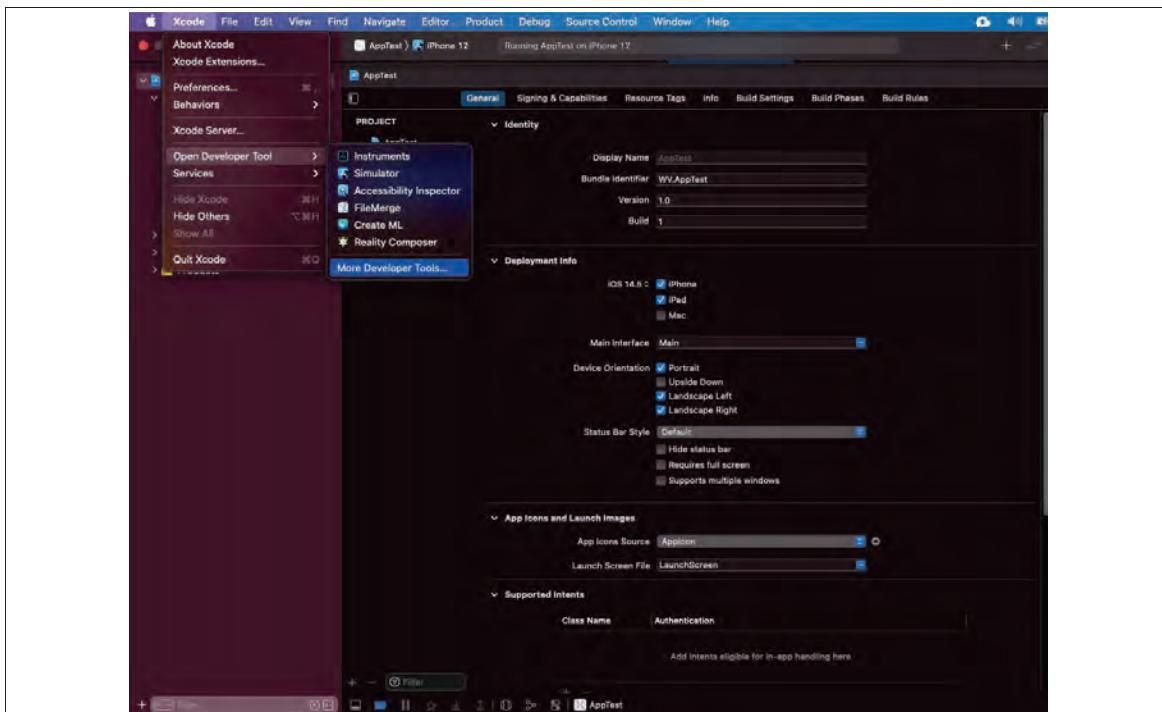


2. Network Link Conditioner on MacOS Computer/Xcode

Network Link Conditioner can also be installed on a computer running Apple’s MacOS, as shown below.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

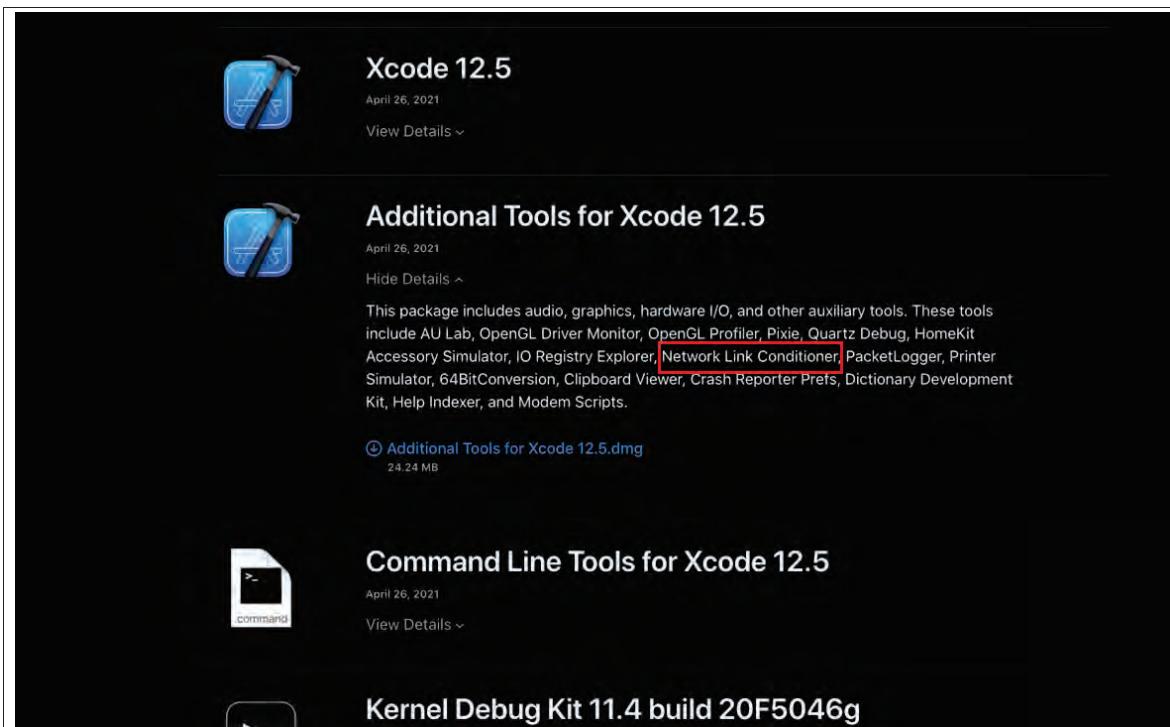


The above image shows the Xcode IDE. A user can access additional Developer Tools by selecting Xcode->Open Developer Tool->More Developer Tools...

Selecting “More Developer Tools...” opens a web browser to a web page where Additional Tools (including Network Link Conditioner) can be downloaded:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

Network Link Conditioner is installed to (and executed from) the System Preferences preference pane in MacOS:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1pre] A system for testing an application for a mobile device comprising:

The above image shows the System Preference pane on a MacOS computer. The Network Link Conditioner can be reached by selecting the icon in the annotated red box.

On information and belief, Defendant uses Apple's Xcode Developer Tools to test mobile banking applications for its business, for example—Chase Mobile. While Chase Mobile is identified as an example application, the contentions detailed in this chart apply to all mobile application testing done by or on behalf of Defendant using the Accused Instrumentalities. On information and belief, Defendant's testing of mobile banking applications includes using the features detailed throughout this document.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

The Accused Instrumentalities, including Apple's Xcode Developer Tools, are configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application.

Discussion of this limitation is broken into two sections: (1) provides an overview of how Xcode Developer Tools simulate/emulate characteristics of devices and networks; and (2) provides an overview of how Xcode Developer Tools display resources available to or utilized by applications.

1. Xcode Developer Tools Emulation/Simulation Overview

When an Xcode user writes an application, the user can choose to execute that application on either a physical device, or on a simulator by selecting the destination from within Xcode IDE:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Select a Simulated Device

For iOS, tvOS, and watchOS apps, you can choose a simulated device, under [Platform] Simulators, from the *run destination menu* next to the scheme menu in the toolbar.

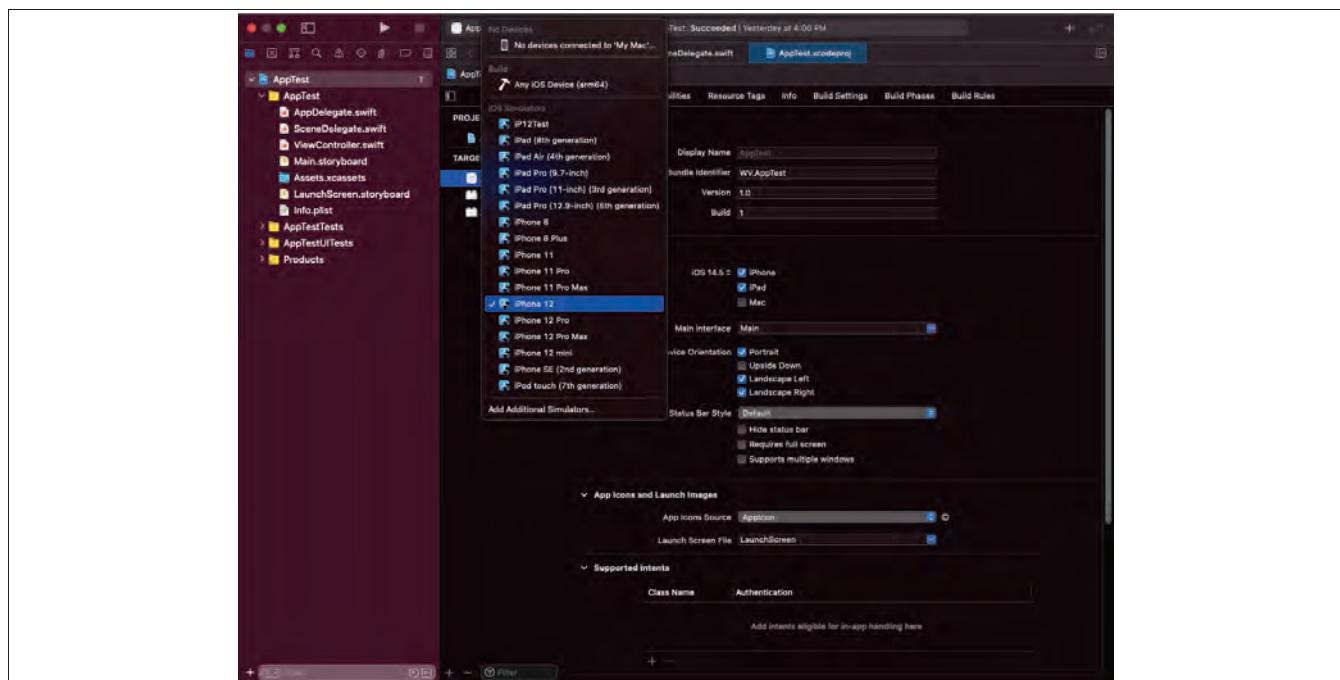
To add additional simulators of a product family running older versions of the operating system, choose Add Additional Simulators.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

<https://developer.apple.com/documentation/xcode/running-your-app-in-the-simulator-or-on-a-device>, accessed on December 28, 2021.

Xcode displays a list of iOS mobile device models (such as iPhones and iPads) from which a user can select to simulate how the application will run on that model. The application being tested can then be deployed on a model that is specific to that device.



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Each model includes characteristics indicative of a corresponding mobile device, such as screen size, resolution, and button availability.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

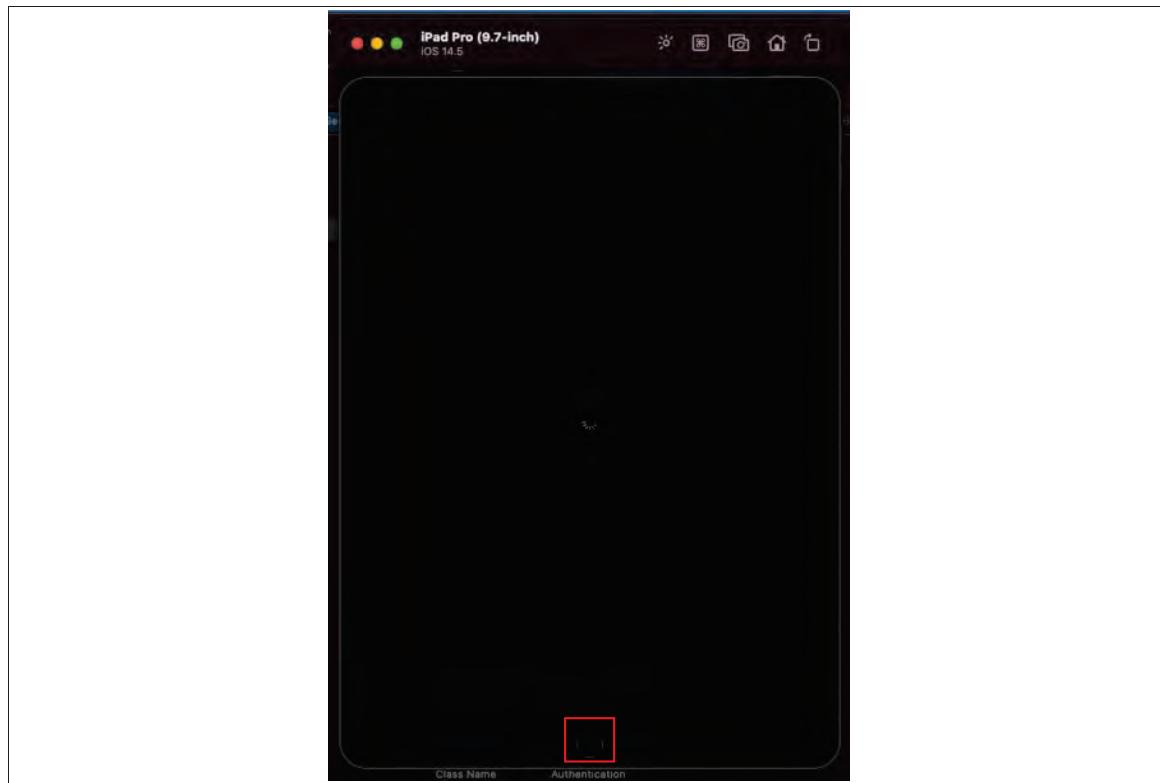
Simulated iPhone 12:



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

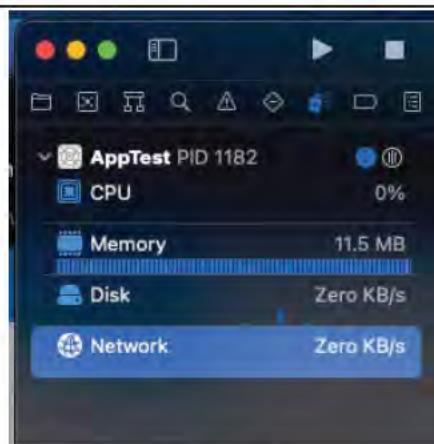
Simulated iPad Pro (button in box in red):



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Xcode Simulator can also be used to simulate other mobile device characteristics, such as the CPU, Memory, and Disk Storage that would be available to the application if it were running on a physical device. For example, the screenshot below shows resource usage and availability of an application called “AppTest” as it is being run on a simulated iPhone 12:



These characteristics are shown for both simulated and physical devices.

Network Link Conditioner (in combination with either Xcode Simulator or a physical device) can be used to emulate/simulate network characteristics indicative of performance of the mobile device when executing the application.

Emulated network characteristics include, *inter alia*, in/out bandwidth, in/out packet loss, in/out delay, and DNS delay. These network characteristics can be set in Network Link Conditioner either on a physical iOS device, or in the Network Link Conditioner in MacOS. For example, by setting bandwidth limits through Network Link Conditioner, a developer can limit the bandwidth available to be used by the application being executed.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



As shown below, Network Link Conditioner settings can be set on both (i) a physical iOS device, as well as (ii) a MacOS device (such as a computer). Setting the Network Link Conditioner settings on a MacOS device simulates/emulates the network characteristics available to be used by applications running on Xcode Simulator on that same MacOS device.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

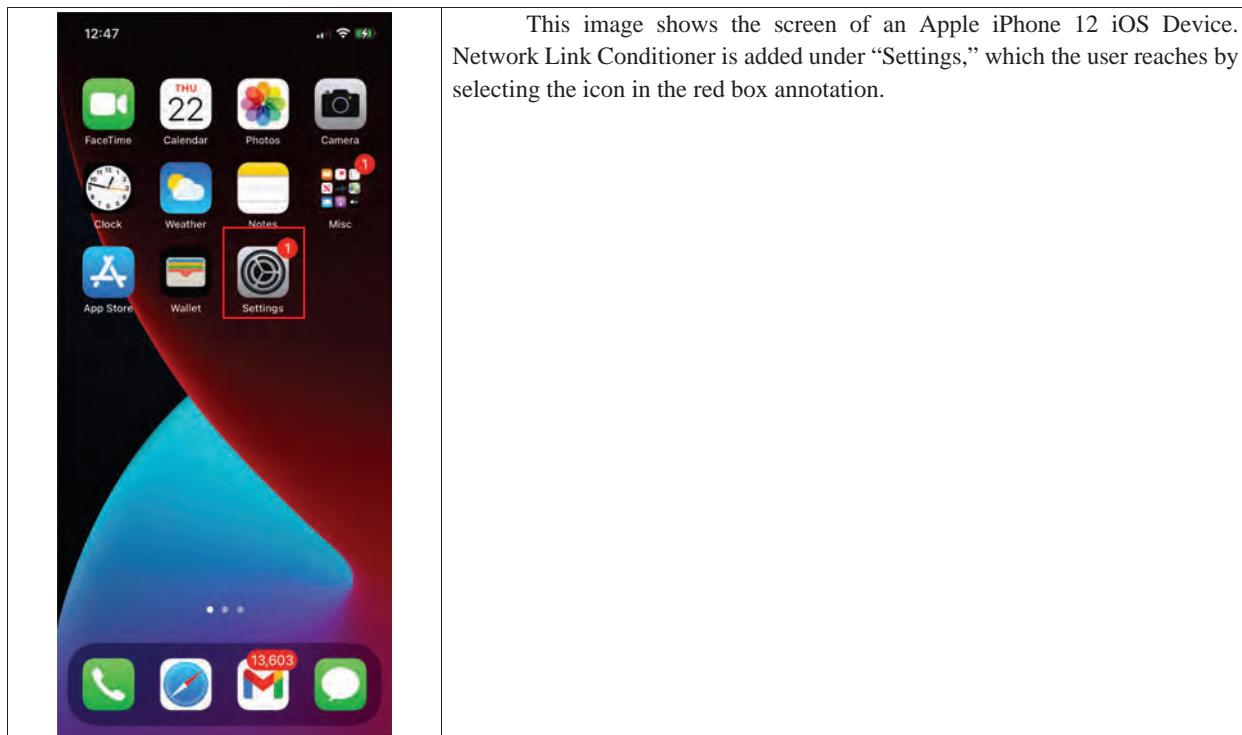
By setting the emulated/simulated network values in Network Link Conditioner, a developer can limit, for example, maximum available bandwidth into or out of the device. After setting these limits, a developer will know what bandwidth is available to the device, for example by comparing current bandwidth usage against the maximum bandwidth available.

a. *Network Link Conditioner on iOS*

The following steps are performed on an iOS Device, such as an iPhone 12. Note that the use of a specific model or version device is merely exemplary, and not meant to be limiting.

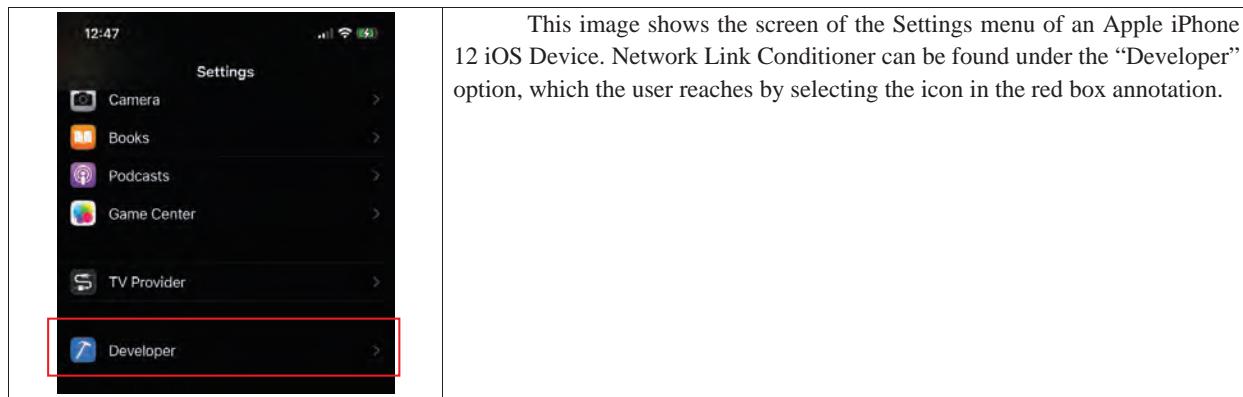
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



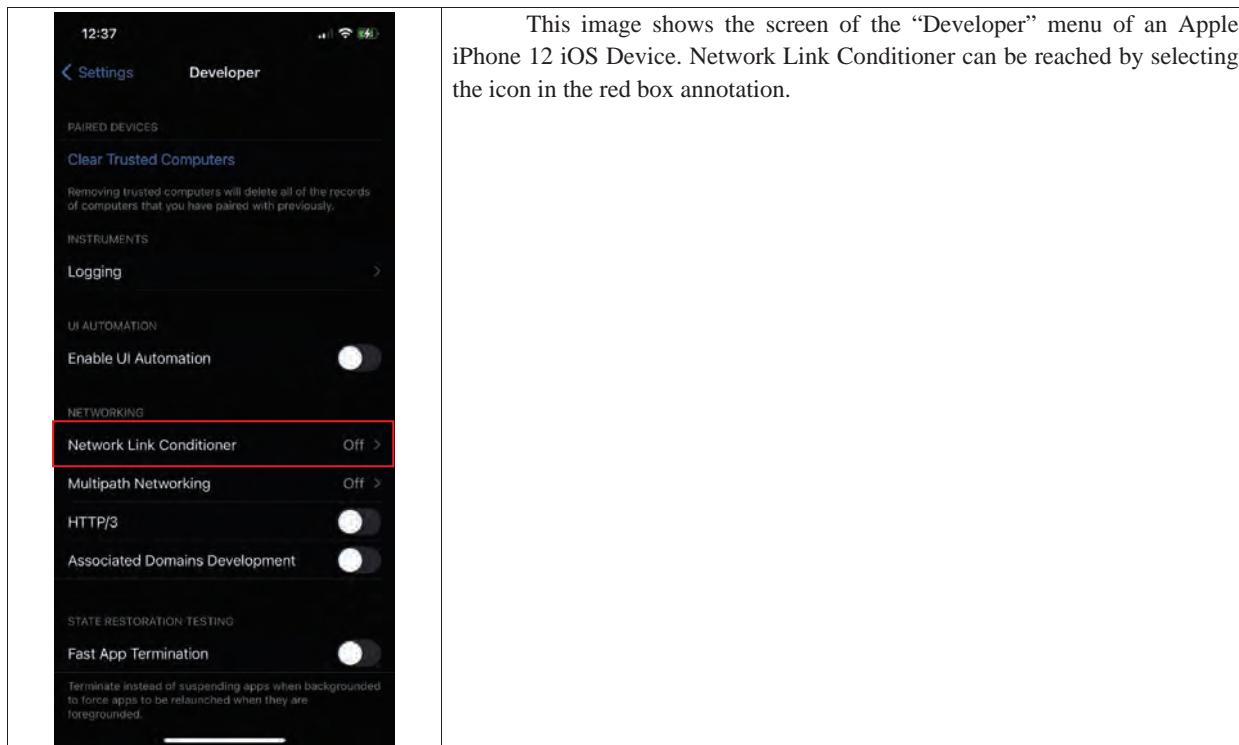
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



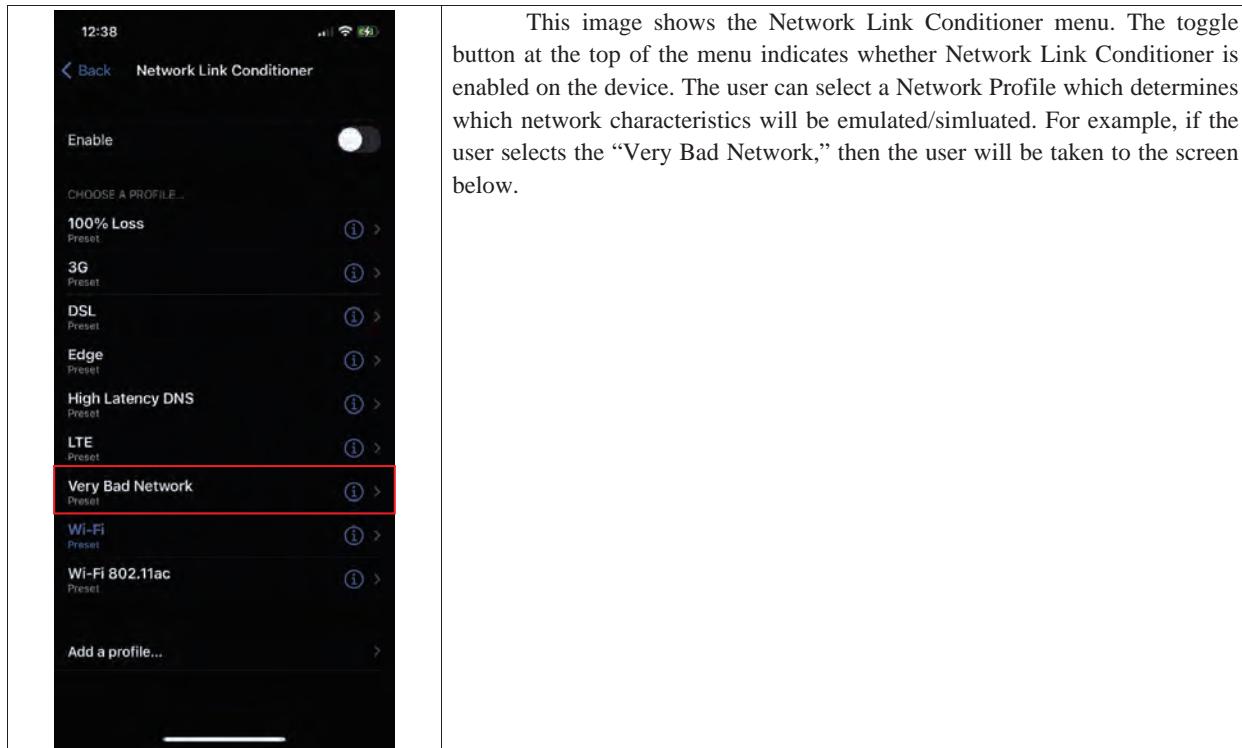
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



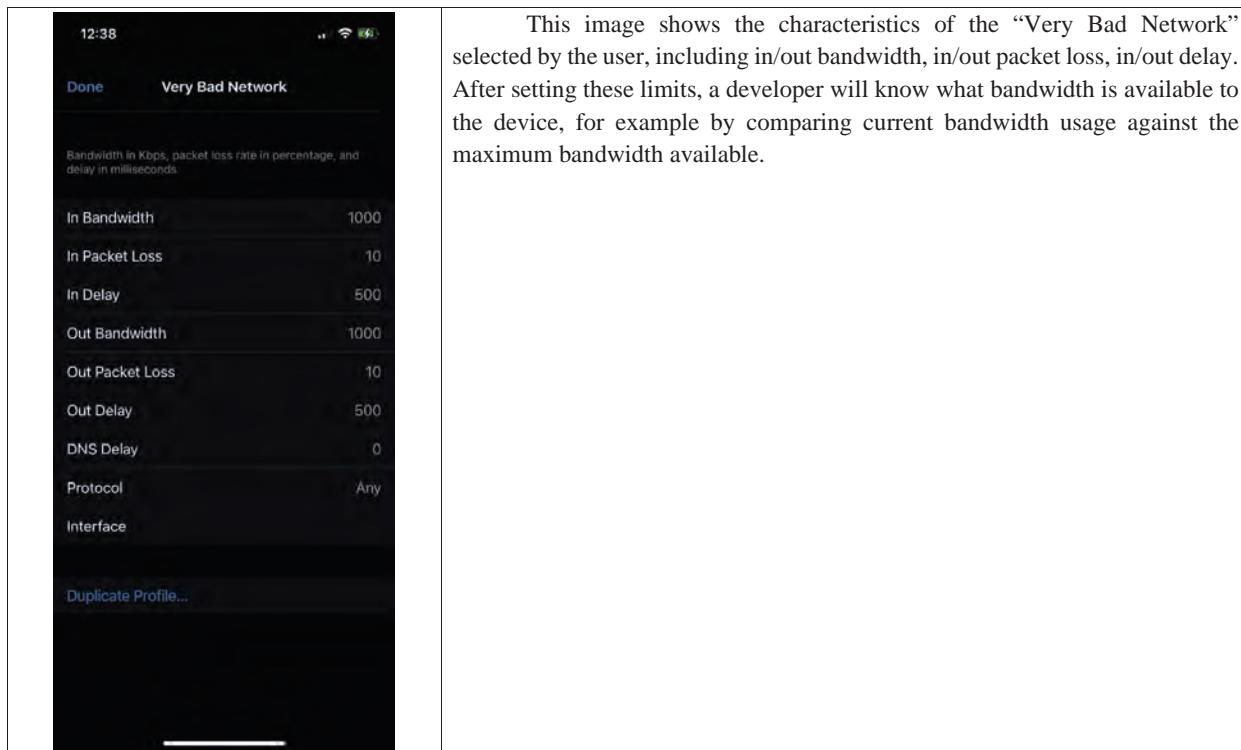
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

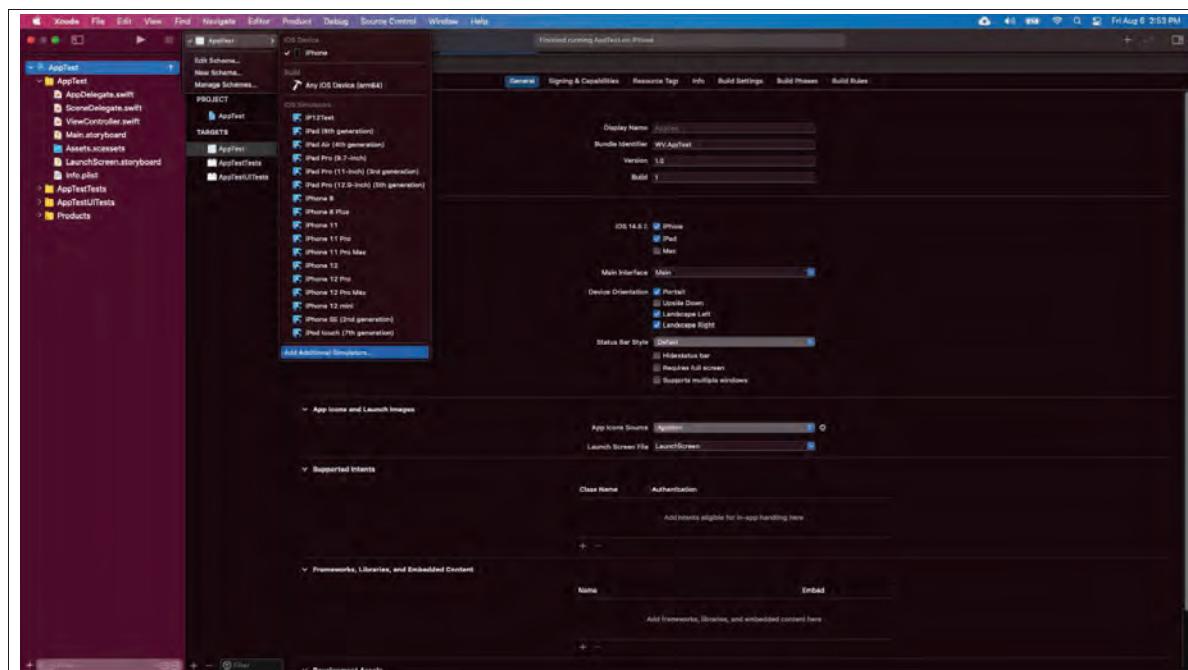
[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Link Conditioner settings can also be set for the device directly through Xcode IDE. While these settings are made through Xcode IDE, they will be effective on the iOS device connected to the MacOS computer:

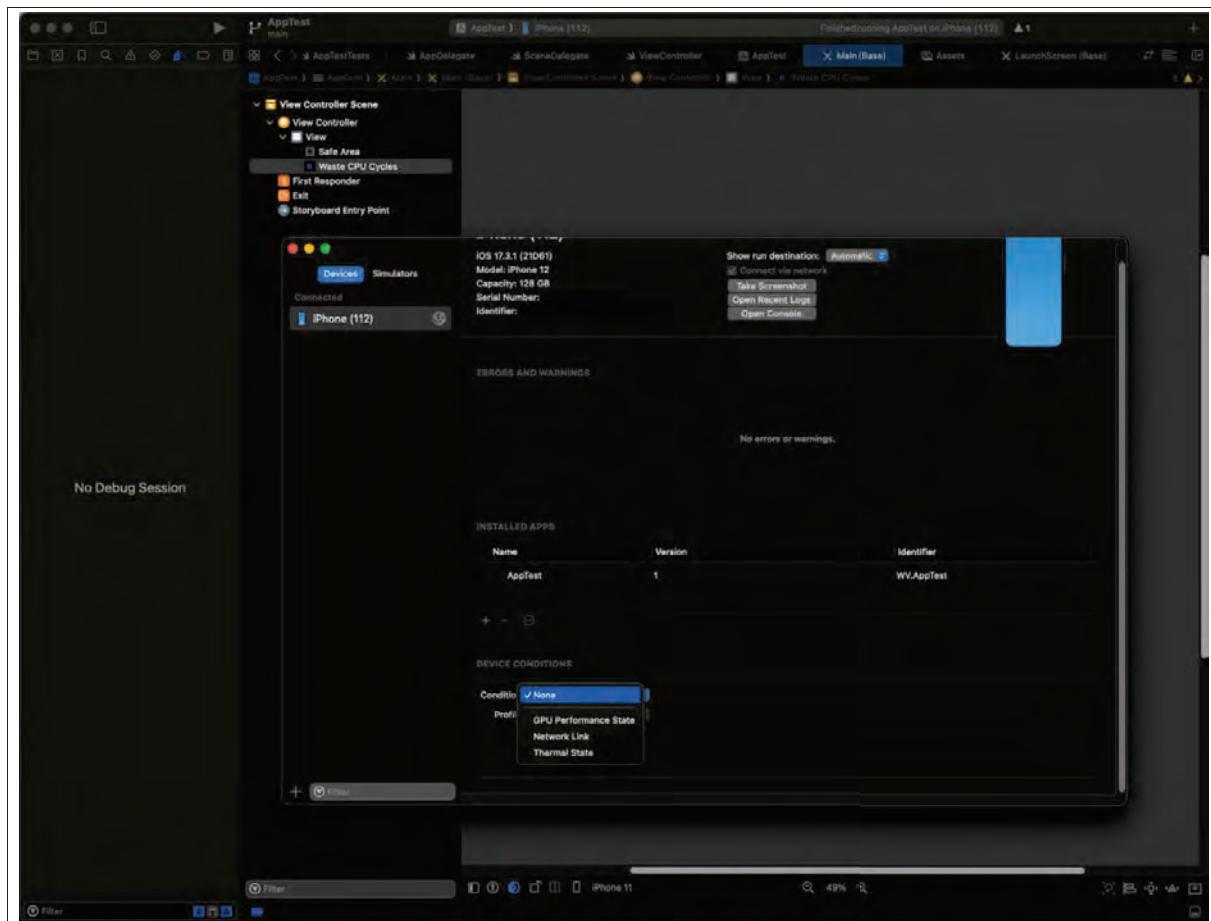


Users can access the Link Conditioner settings from Xcode IDE by selecting the “Add Additional Simulators...” menu setting.

By selecting the “Devices” option, users can set Device conditions.

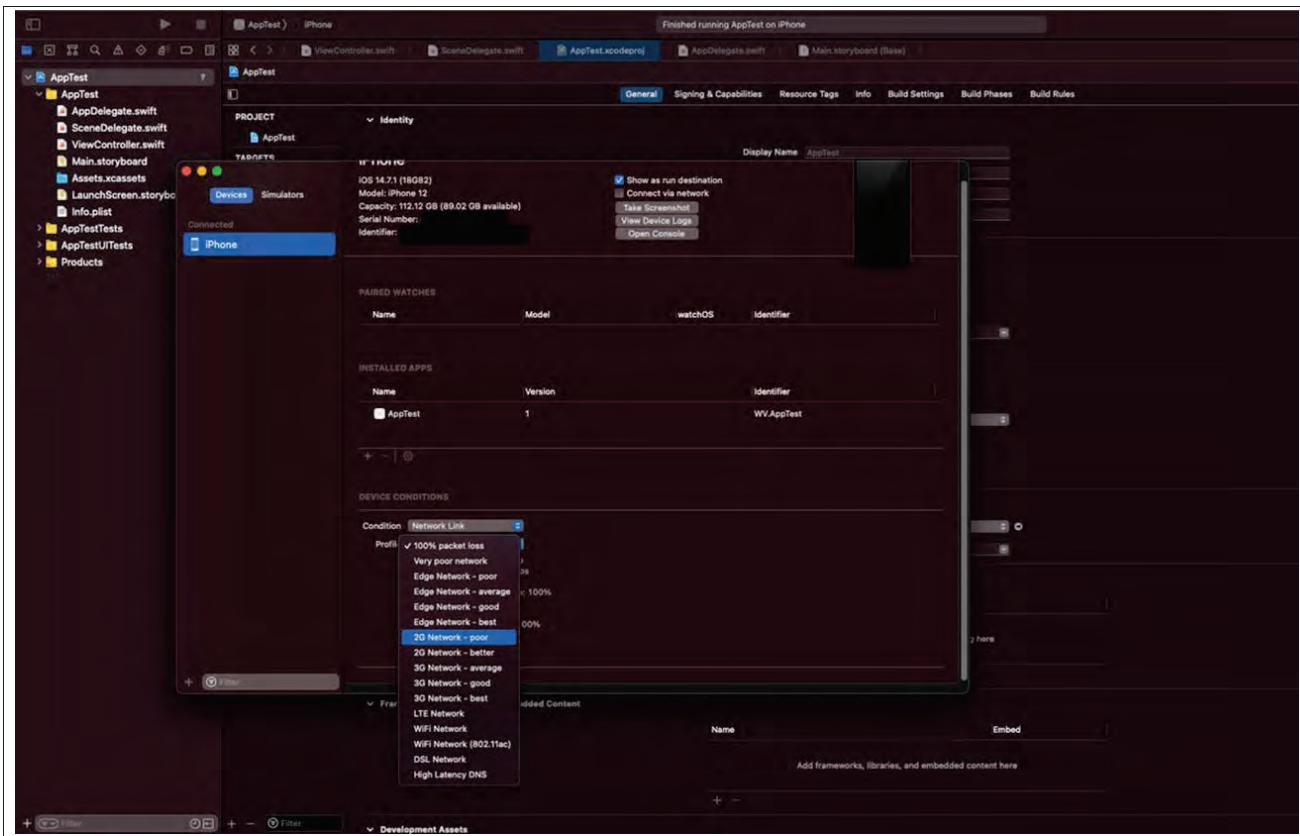
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



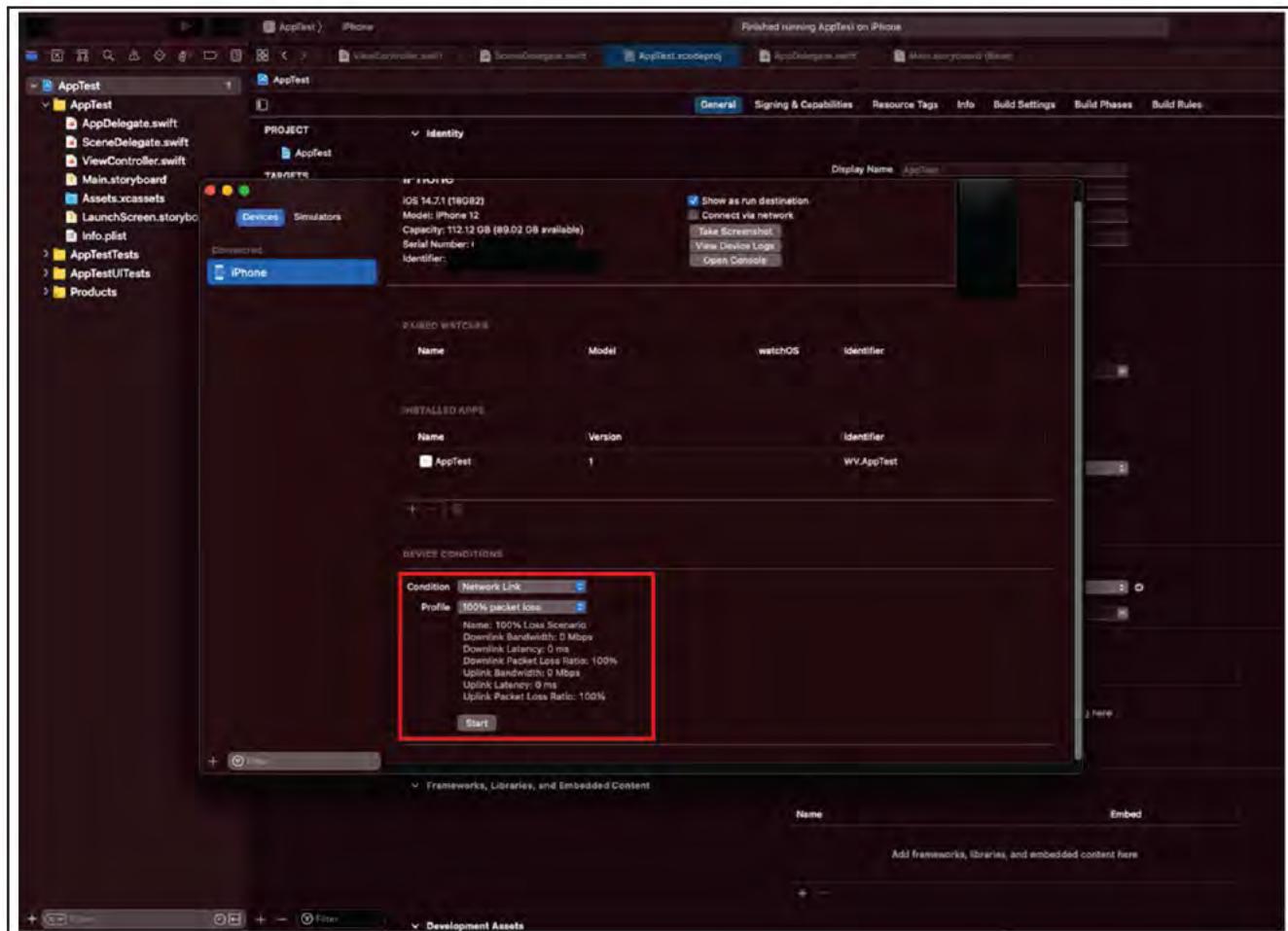
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

If users select “Network Link” in the first dropdown box, the user can then select a profile representing network characteristics in the second dropdown box.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

As shown above, after selecting the “100% packet loss” profile, the characteristics (including bandwidth, latency, and packet loss) are selected based on that profile.

a. *Network Link Conditioner on MacOS*

Users can also emulate/simulate network characteristics indicative of performance of the mobile device when executing the application by using the Network Link Conditioner tool from System Preferences in MacOS.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Users select the Network Link Conditioner (shown in red) from System Preferences in MacOS.



From Network Link Conditioner, users can select a profile (such as “100% Loss”, “3G”, etc.) which simulates/emulates network characteristics.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Simulated network characteristics include bandwidth, packets dropped, and delay. After setting these values, a developer can determine the resources available to the application by comparing the current resource utilization (such as bandwidth) against the values set in Network Link Conditioner.

2. Xcode Developer Tool Display Overview

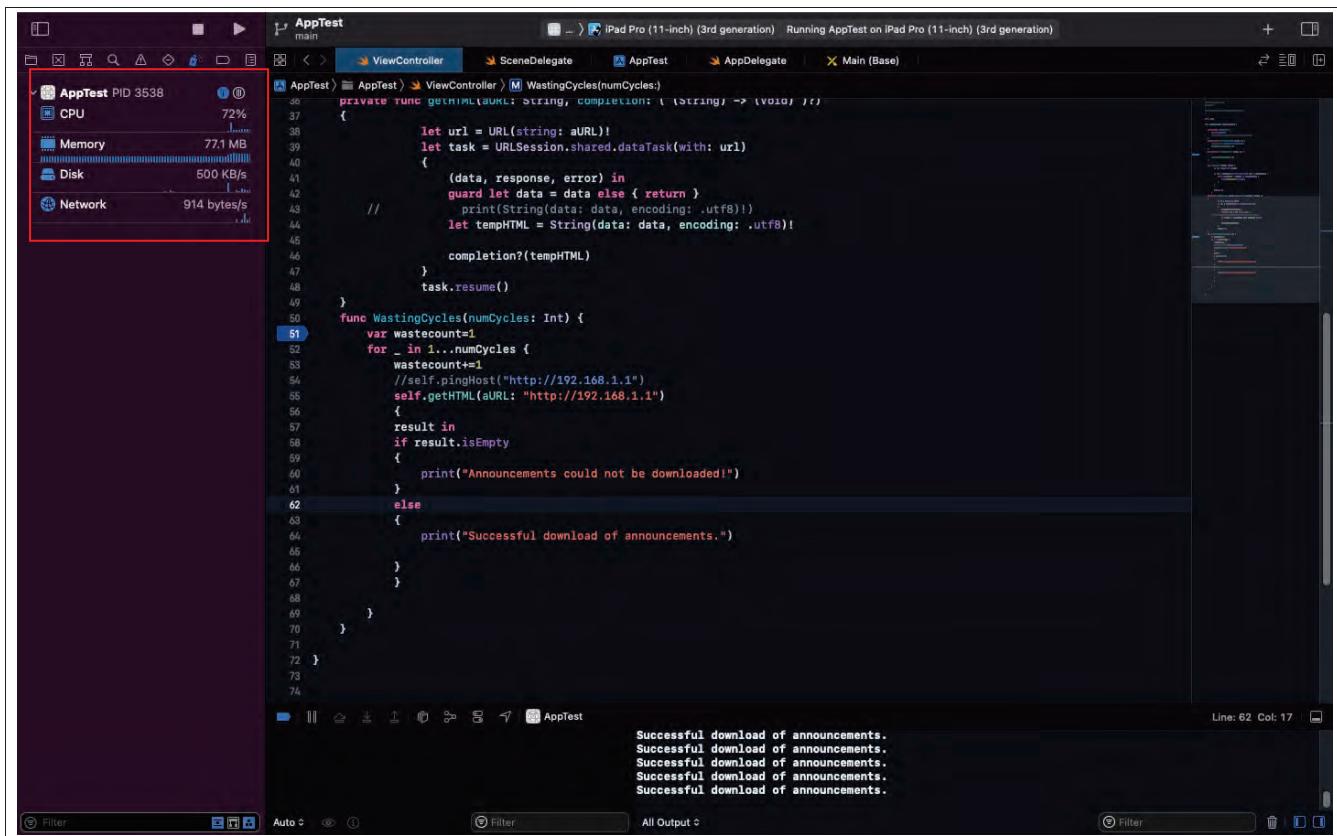
Apple's Xcode Developer Tools are used to visually display the resources available to, and used by, the application. These resources are displayed through tools such as (a) Xcode IDE and (b) Xcode Instruments.

a. Xcode IDE

When a developer runs the application being tested through Xcode IDE, then Xcode IDE displays a plurality of visual representations of the resources available to (and being used by) the application, with each bar in the bar graph being displayed at the same time and indicating the resources being used and that are available:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



The screenshot shows the Xcode interface. On the left, the 'Activity Monitor' window is open, displaying resource usage for the process 'AppTest' (PID 3538). It shows CPU at 72%, Memory at 7.71 MB, Disk at 500 KB/s, and Network at 914 bytes/s. A red box highlights this window. To the right, the main Xcode window displays the 'AppTest' code. The code contains a function 'WastingCycles' that performs a loop to download files from a local host. The output pane at the bottom shows multiple lines of text: 'Successful download of announcements.', repeated five times.

```

private func getHTML(from: String, completion: @escaping (String) -> Void) {
    let url = URL(string: aURL)!
    let task = URLSession.shared.dataTask(with: url) {
        (data, response, error) in
        guard let data = data else { return }
        print(String(data: data, encoding: .utf8)!)
        let tempHTML = String(data: data, encoding: .utf8)!

        completion?(tempHTML)
    }
    task.resume()
}

func WastingCycles(numCycles: Int) {
    var wasteCount=1
    for _ in 1...numCycles {
        wasteCount+=1
        //self.pingHost("http://192.168.1.1")
        self.getHTML(aURL: "http://192.168.1.1")
    }
    result in
    if result.isEmpty {
        print("Announcements could not be downloaded!")
    } else {
        print("Successful download of announcements.")
    }
}
}

```

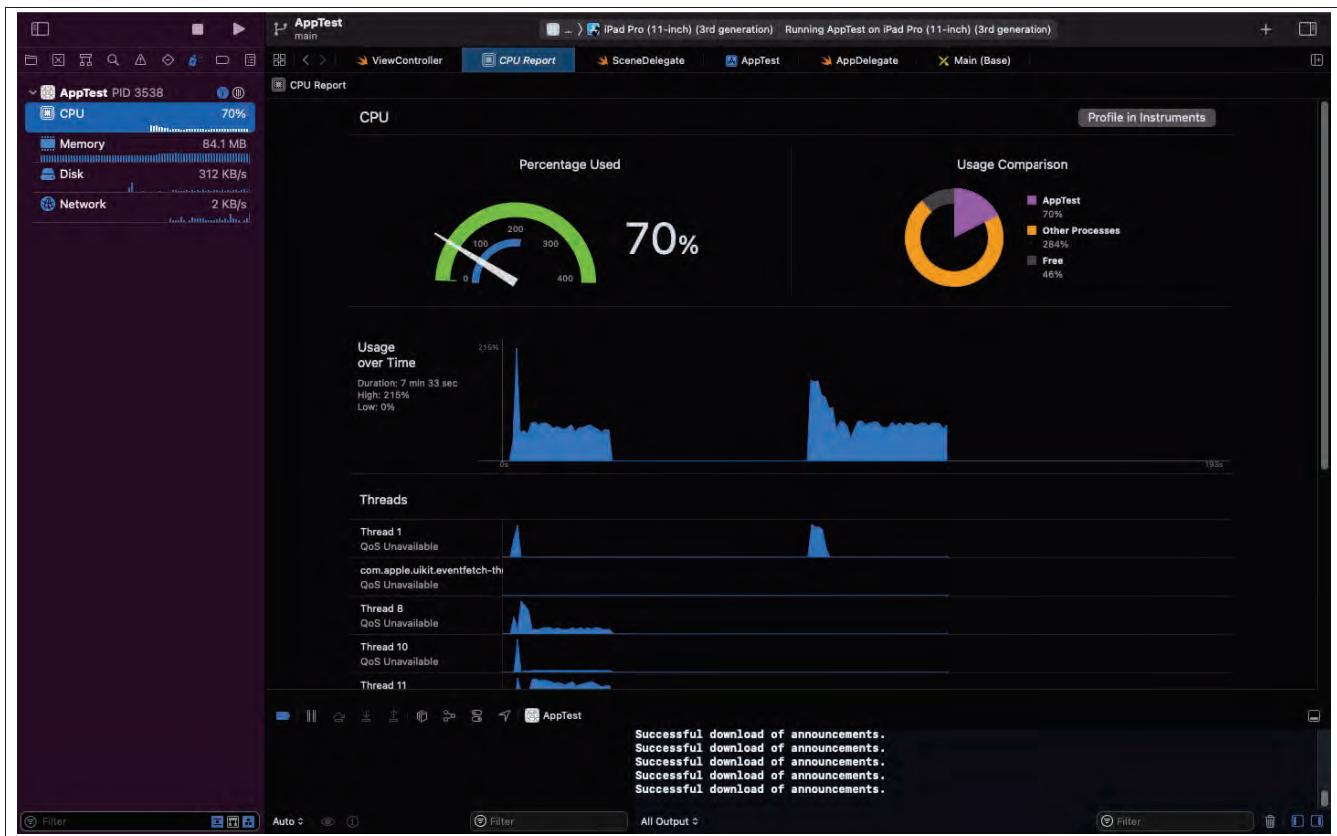
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Developers can view additional visual representations of the resources used and available by clicking on each individual graph. For example, clicking CPU shows the following (including the percentage currently used (and the percentage available, e.g., 100% minus the percentage being used), usage over time, usage by thread, and a comparison of the application's CPU usage vs other processes):

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



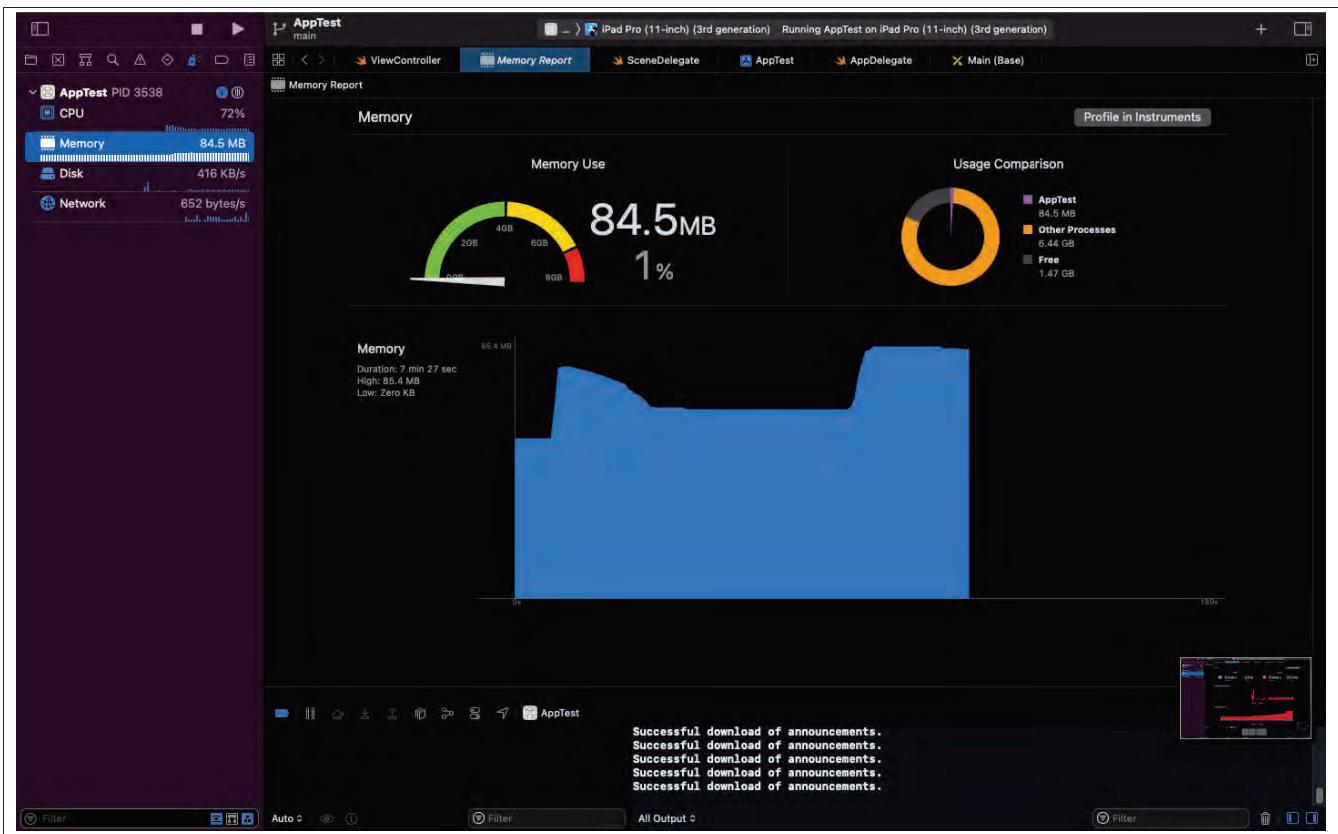
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Similarly, clicking Memory provides a visual display including graphs showing memory currently used, total memory available, % currently used, usage comparison of the application vs other processes, and memory usage over time. Furthermore, developers can compare the memory currently used against the memory available in the physical device:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



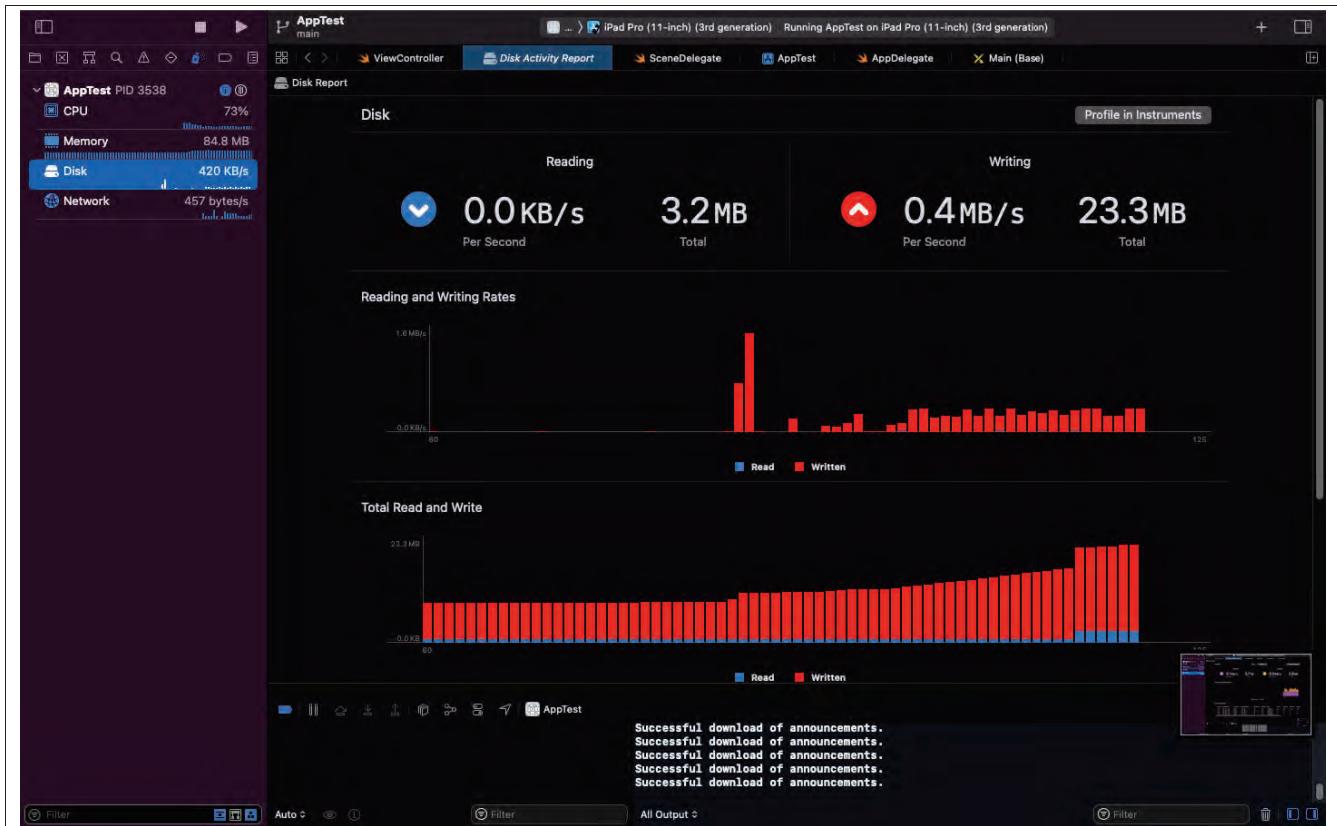
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Disk Usage (showing current bytes read/written per second, total bytes read or written, reading and writing rates over time, and total reads and writes over time) Furthermore, developers can compare the disk characteristics currently used against the disk characteristics in the physical device:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



This screen shows current disk usage, as well as the disk usage resources still available. Furthermore, a developer could use known maximum disk read/write rates to determine available resources based on utilized resources.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

The images below show Network Activity. Network Activity is shown in terms of either Bytes or Packets, depending on which option is selected. A developer could use the limitations selected in Network Link Conditioner (such as maximum bandwidth rates) to determine available resources based on utilized resources.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

The Bytes option is shown below (displaying current sending and receiving rates, total data sent and received, and sending and receiving rate speed over time):



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



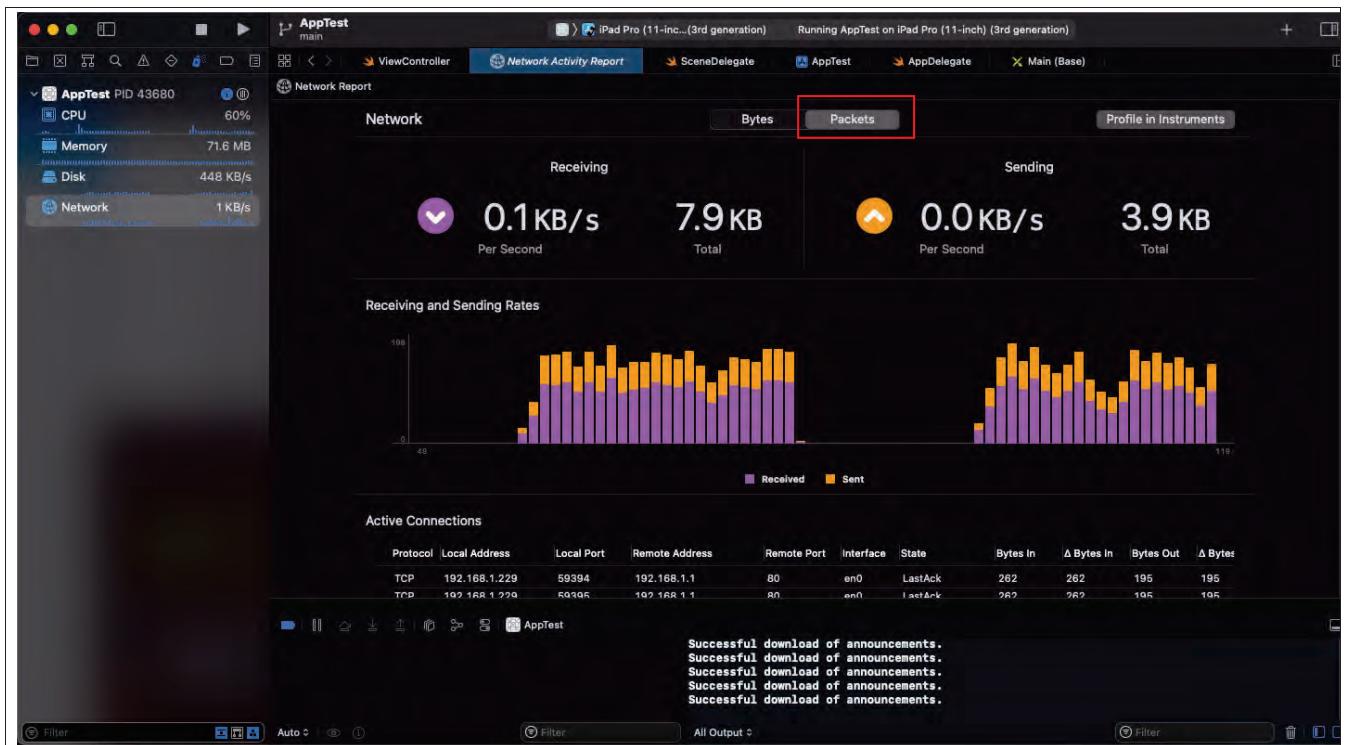
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

The Packets option is shown below (displaying current sending and receiving rates, total data sent and received, and sending and number of packets sent or received over time). The timing and quantity of packets sent correspond to the network latency and packet loss.

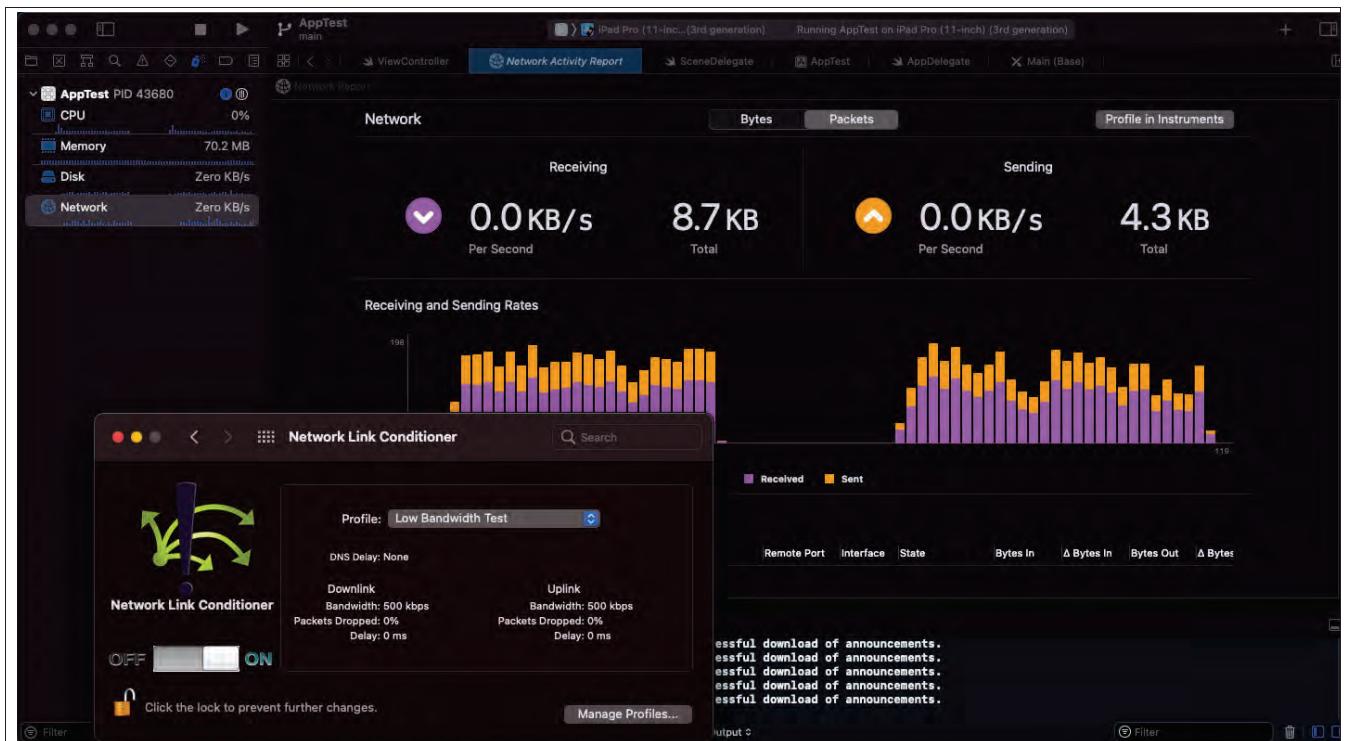
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

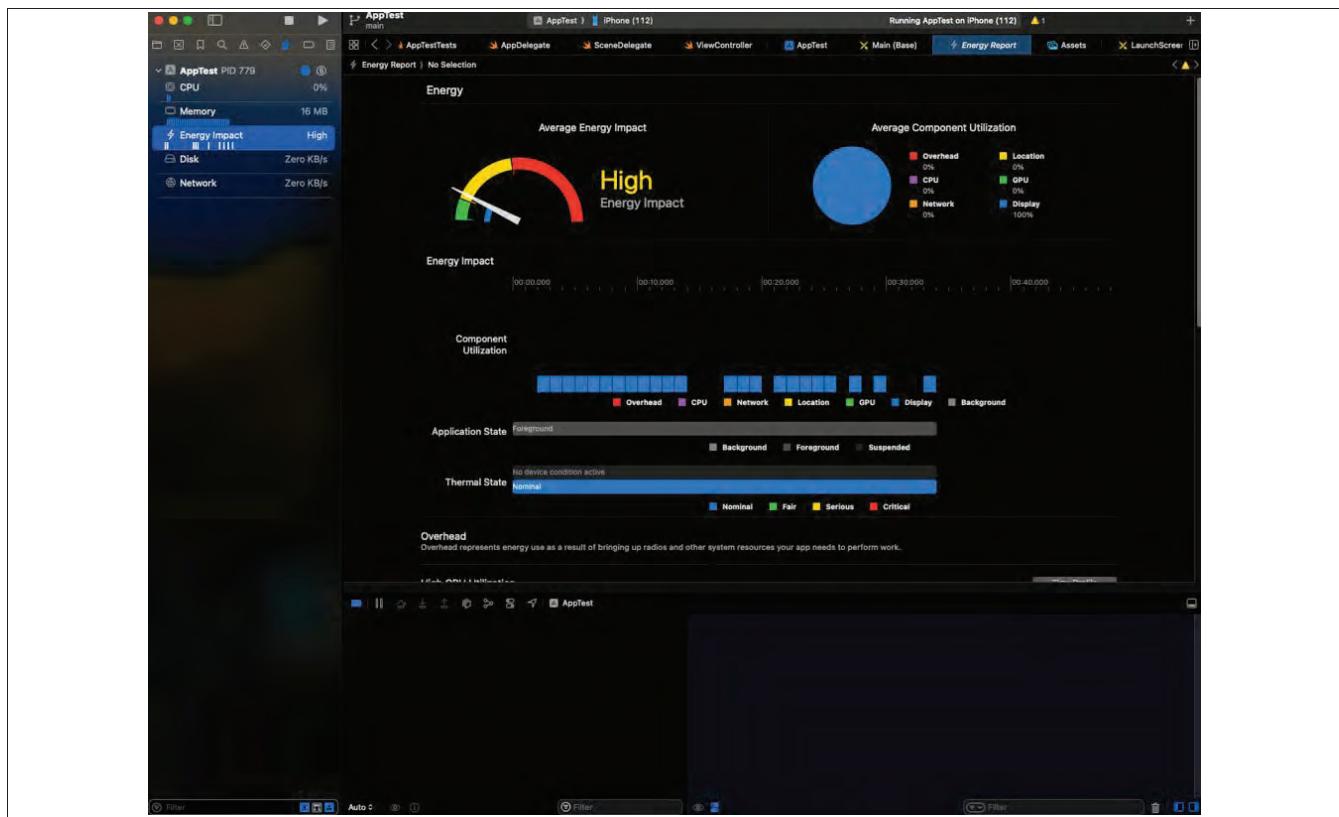
[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

The Energy Impact option shown below shows current energy impact of the application, as well as energy still available.



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

b. Xcode Instruments

The Xcode Instruments tool is also used to visually display the resources available to, and used by, the application. As described by Apple, the “Instruments” tool:

Instruments is a powerful and flexible performance-analysis and testing tool that's part of the Xcode tool set. It's designed to help you profile your iOS, watchOS, tvOS, and macOS apps, processes, and devices in order to better understand and optimize their behavior and performance. Incorporating Instruments into your workflow from the beginning of the app development process can save you time later by helping you find issues early in the development cycle.

In Instruments, you use specialized tools, known as *instruments*, to trace different aspects of your apps, processes, and devices over time. Instruments collects data as it profiles, and presents the results to you in detail for analysis.

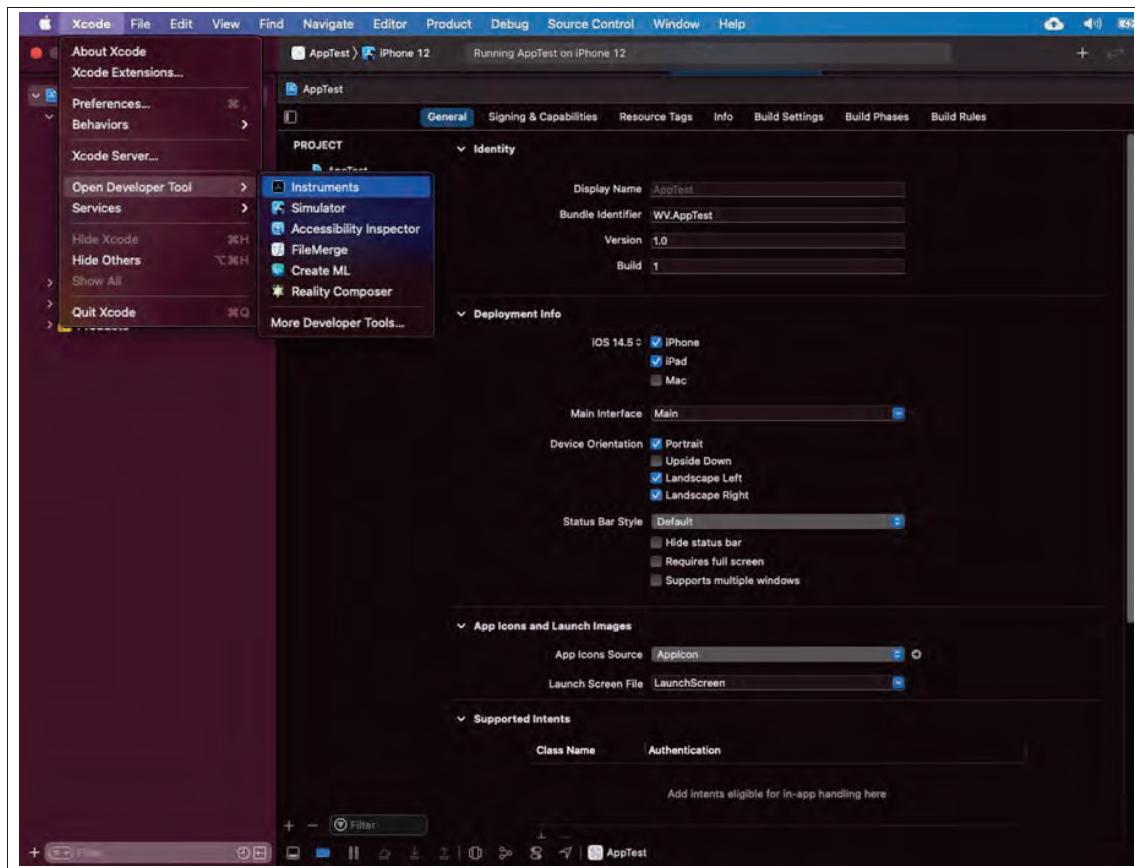
Unlike other performance and debugging tools, Instruments allows you to gather widely disparate types of data and view them side by side. This makes it easier to identify trends that might otherwise be overlooked. For example, your app may exhibit large memory growth caused by multiple open network connections. By using the Allocations and Connections instruments together, you can identify connections that are not closing and thus resulting in rapid memory growth.

<https://help.apple.com/instruments/mac/current/#/dev7b09c84f5>, accessed on March 12, 2024.

To access Xcode Instruments, an Xcode IDE user may select the “Instruments” menu option from Xcode’s menu:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

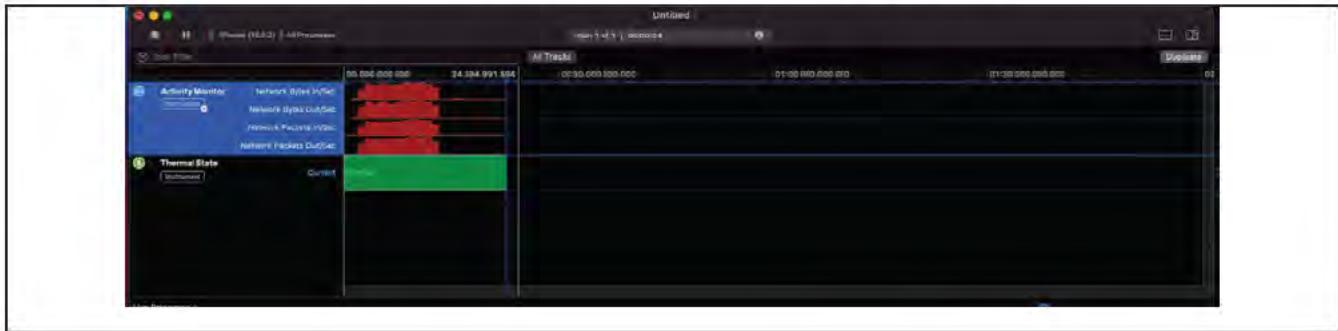
Multiple Instrument templates are available related to various resources, as shown in the examples below:



The Instruments window displays one or more windows showing simultaneous displays of resources of the mobile device that are used by and/or available to the application currently running;

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Resources displayed in instruments include those related to CPU (Total Load, User Load, System Load), Thermal State, Memory (Memory Used/Available (for example, the total memory minus the memory used), Cached Files, Compressed Memory, Swap Used), Disk (Bytes Read or Written/Second, Disk Read or Write Operations/Second), Network interactions (Network Bytes In or Out/Second, Network Packets In or Out/Second), GPU Usage, and others:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1a] software configured to simulate, via one or more profile display windows, a plurality of network characteristics indicative of performance of the mobile device when executing the application;

Xcode Instruments users may select what they wish to see, and multiple graphs may be displayed at once. The size of the bar in each bar graph shows both the resources utilized, as well as the resources available to the application.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.

The network characteristics simulated by Xcode Network Link Conditioner are based on data of interaction with networks in non-simulated environments.



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

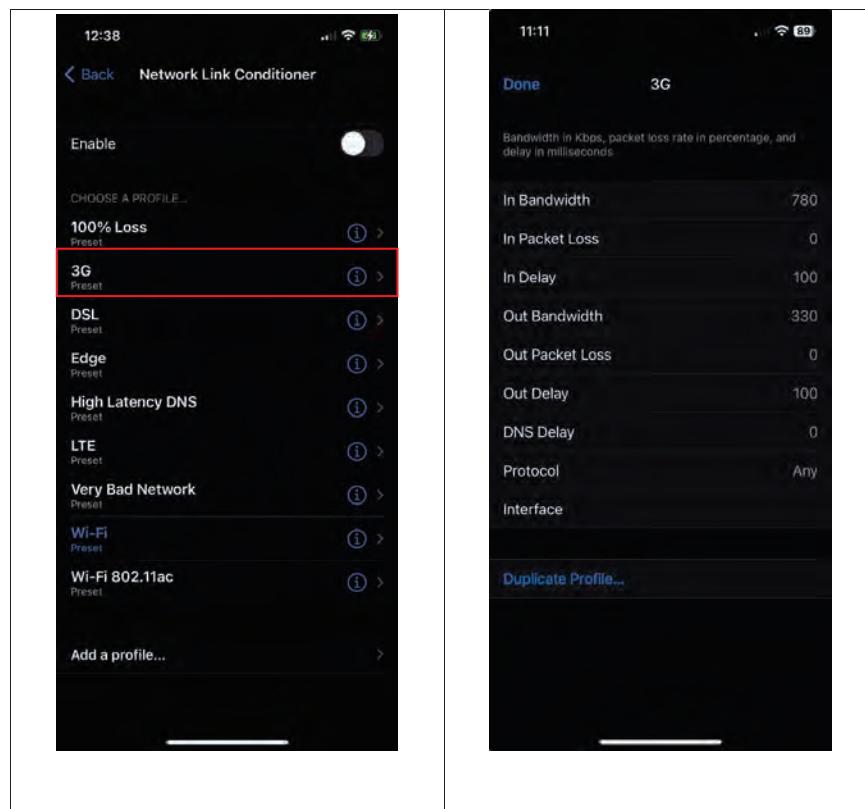
[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.



Network Link Conditioner on MacOS.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

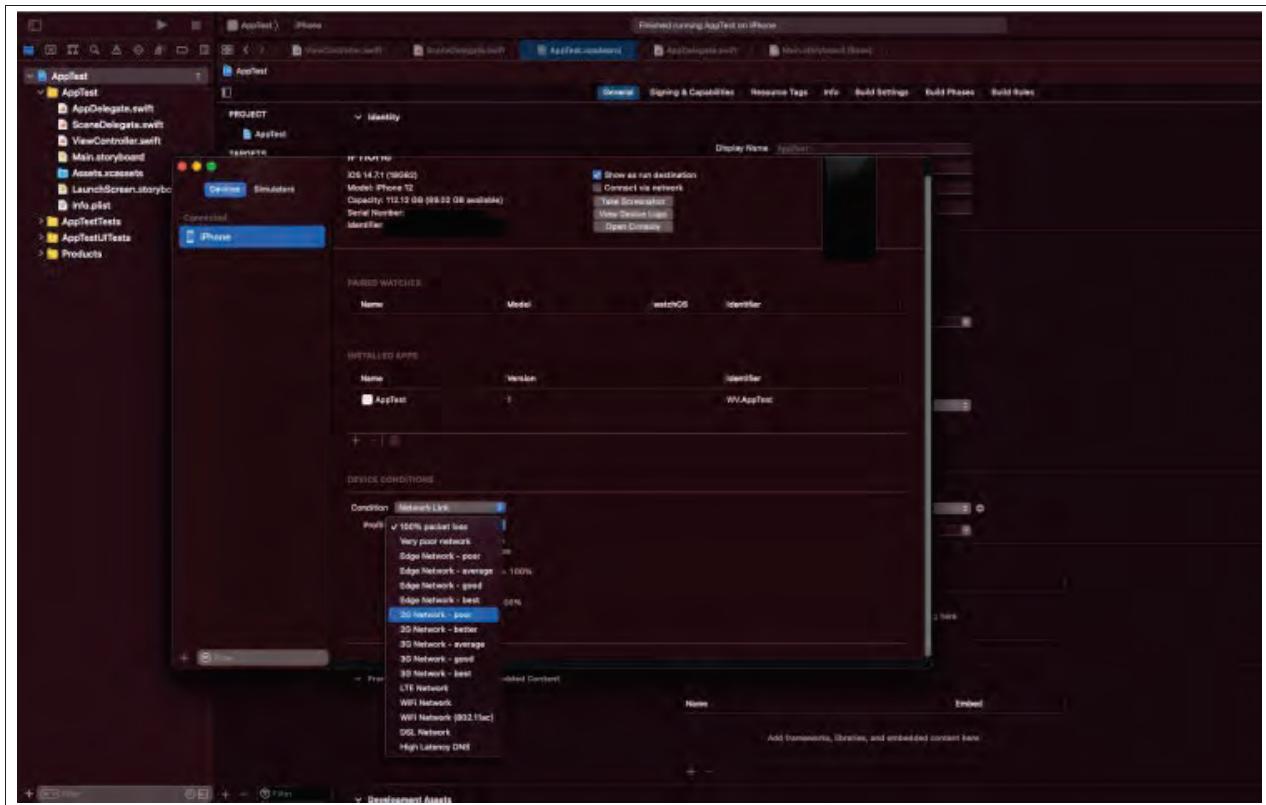
[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.



Network Link Conditioner on iOS.

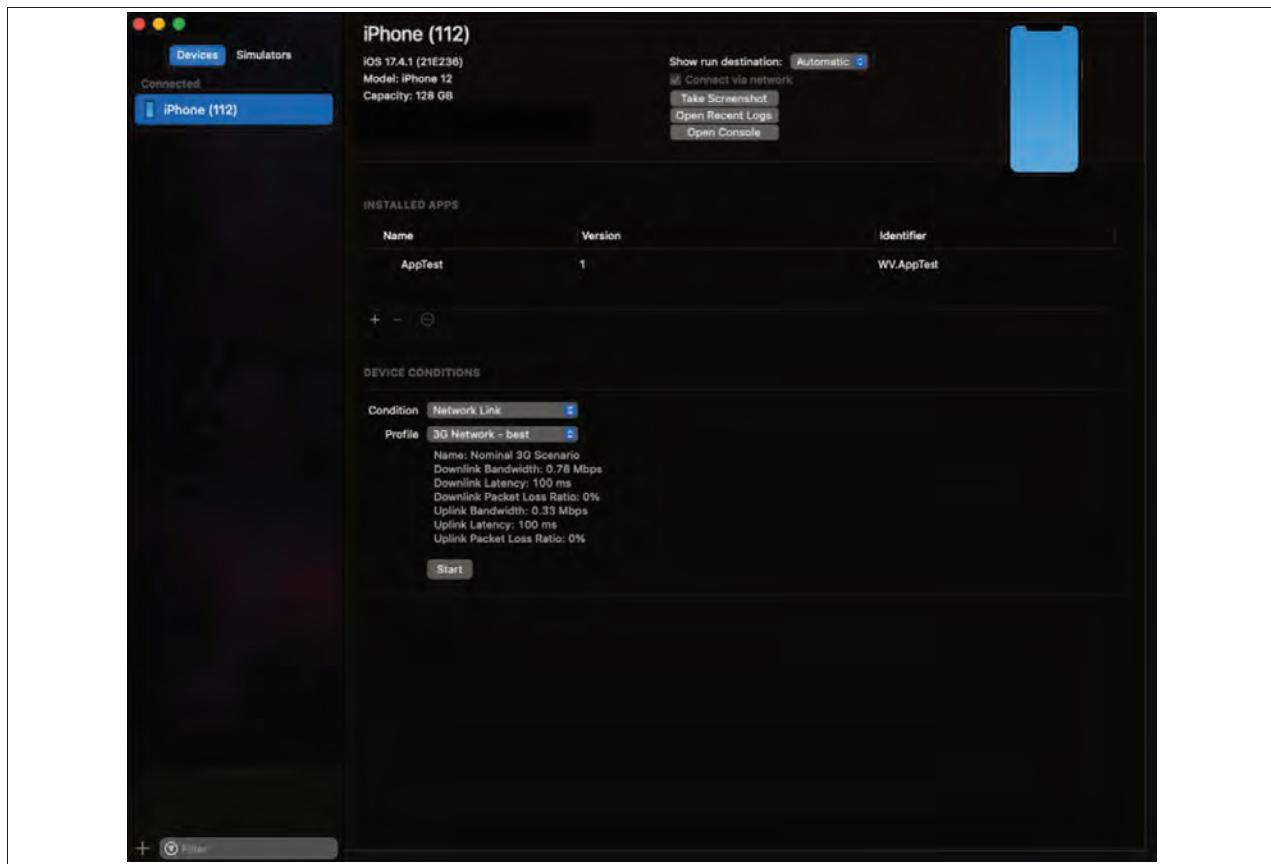
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.



Network Link Conditioner on iOS configured through Xcode IDE.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.

The network characteristics such as bandwidth, latency/delay, and packet loss are based on data of interactions with networks in non-simulated environments at least because of their reliance on network standards developed to dictate the operation of real-world networks, such as LTE, 2G, 3G, and EDGE.

For instance, GSM corresponds to what is often referred to as the “2G” network standard established around 1991. Similarly, EDGE corresponds to Enhanced Data rates for GSM Evolution, an enhancement to GSM. And HSCSD corresponds to High Speed Circuit Switched Data, which is an enhancement to the data rate of circuit switched data in a GSM network. GPRS corresponds to a packet-oriented enhancement to 2G networks—General Packet Radio Service. UMTS is the Universal Mobile Telecommunications System, a new architecture that provided the basis for what is often referred to as the “3G” network standards. HSDPA, or High-Speed Downlink Packet Access, is an enhancement to the 3G network architecture to boost data capacity and improve download rates. Finally, LTE, or Long Term Evolution, represents the transition from 3G to what is typically referred to as “4G” network technology.

Each of these network standards defines the general operation of the network, and these definitions provide theoretical constraints on the networks’ capacity for communication, including bandwidth, latency, packet loss, and speed constraints. These constraints can be further impaired based on network conditions, including the presence of physical obstacles, electro-magnetic interference, and/or distance between the base station and a mobile station with which it is communicating.

The development and evolution of these standards relied on data of interactions with real-world implementations of such networks at least for testing and proof-of-concept. Thus, Network Link Conditioner’s bandwidth, packet loss, and latency/delay constraints correspond to each identified standard, which are based on data of interaction of networks in non-simulated environments.

This is further reinforced by Apple’s documentation, which instructs developers to use Network Link Conditioner to simulate “real-world conditions”:

Test Under Various Conditions

Xcode provides a tool called Network Link Conditioner that can simulate various network conditions, including reduced bandwidth, high latency, DNS delays, packet loss, and so on. Before you ship any software that uses networking, you should install this tool, enable it, then run your software to see how it performs under real-world conditions.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[1b] wherein the network characteristics are based on data of interaction with networks in non-simulated environments.

<https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/WhyNetworkingIsHard/WhyNetworkingIsHard.html>, accessed on March 13, 2024.

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

Claim 8

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

Furthermore, an Xcode user can create one or more scenarios that include scripts. The scenarios in Xcode are testing scenarios used to verify that the application or network is performing adequately.

For example, Xcode users can create unit tests, integration tests, ui tests, and performance tests.

Aim for a “pyramid” distribution of tests, as shown in the figure below. Include a large number of fast, well-isolated unit tests to cover your app’s logic, a smaller number of integration tests to demonstrate that smaller parts are connected together properly, and UI tests to assert the correct behavior of common use cases.

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

In addition to the test pyramid, write performance tests to provide regression coverage of performance-critical regions of code. To learn about the process of identifying performance-critical code, see [Improving your app’s performance](#).

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

UI Tests are specifically intended to “emulate actions of real user behavior.”

UI tests are the ultimate indicator your app works for users in the way you expect, but they take longer to run than other kinds of tests. There are various app variables that can introduce a failure in the same UI test. The test pyramid balances high-fidelity tests that demonstrate that users can complete their tasks, with tightly-focused tests that give you fast feedback about the correctness of your app’s logic and the impact of changes you make.

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

Write a UI test

UI tests work in a different way from unit and integration tests, but they're still organized as methods on subclasses of XCTestCase. Xcode's UI Test Case Class template for new files contains the common starting points for UI tests. Rather than executing your app's code directly, they use the app's user-interface controls as a real user would, to determine whether the user can complete a specific task using the app.

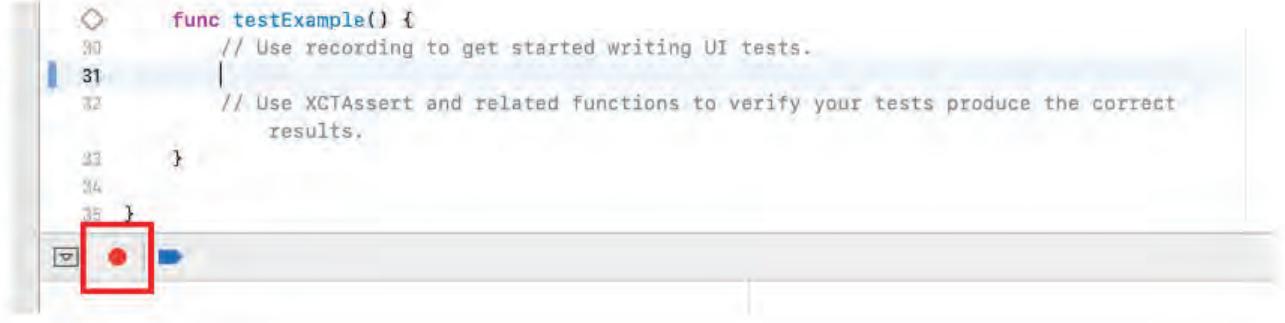
Create UI tests to verify that important user tasks can be completed in the app, and that bugs haven't been introduced that break the behavior of UI controls. UI tests that replicate real user activities provide confidence that the app can be used for its intended task. A UI test for a document-based app might verify that the user can create a new document, edit its content, then delete the document.

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

To create a UI test in a method on an XCTestCase subclass, record your interaction with the app using Xcode's Record UI Test feature. Design UI tests to replicate the most critical workflows that would cause the biggest impact to your users if they broke, and to replay reported bugs so that you can avoid regressions.



```
func testExample() {
    // Use recording to get started writing UI tests.
    //
    // Use XCTAssert and related functions to verify your tests produce the correct
    // results.
}
```

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

Write a performance test

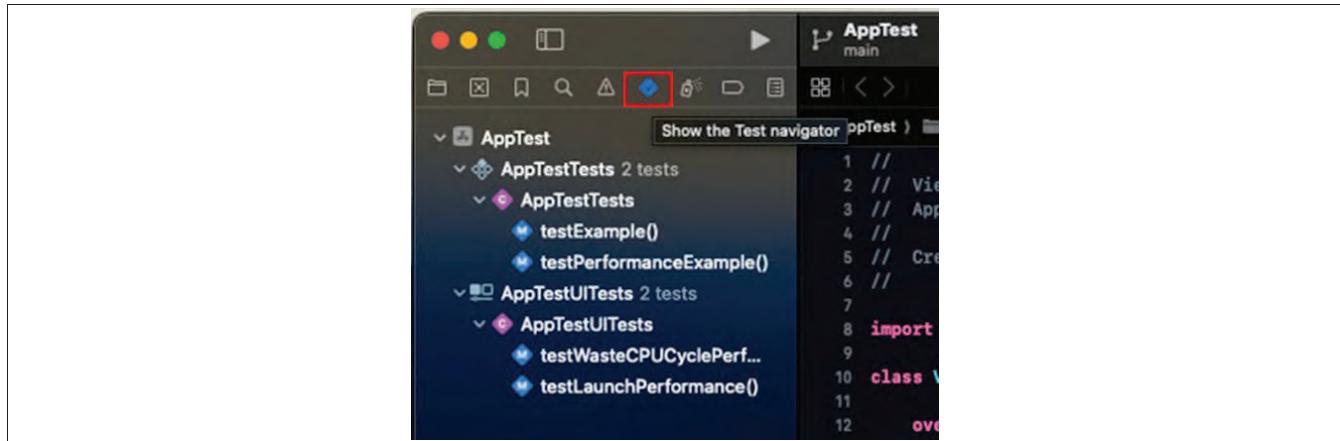
Write performance tests to gather information on time taken, memory used, or data written, during the execution of a region of code. XCTest runs your code multiple times, measuring the requested metrics. You can set a baseline expectation for the metric, and if the measured value is significantly worse than the baseline, XCTest reports a test failure.

<https://developer.apple.com/documentation/Xcode/testing-your-apps-in-Xcode>, accessed on March 15, 2024

Test scripts can be written by accessing the Test Navigator within Xcode IDE, for example by clicking on the icon in the red annotated box below:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

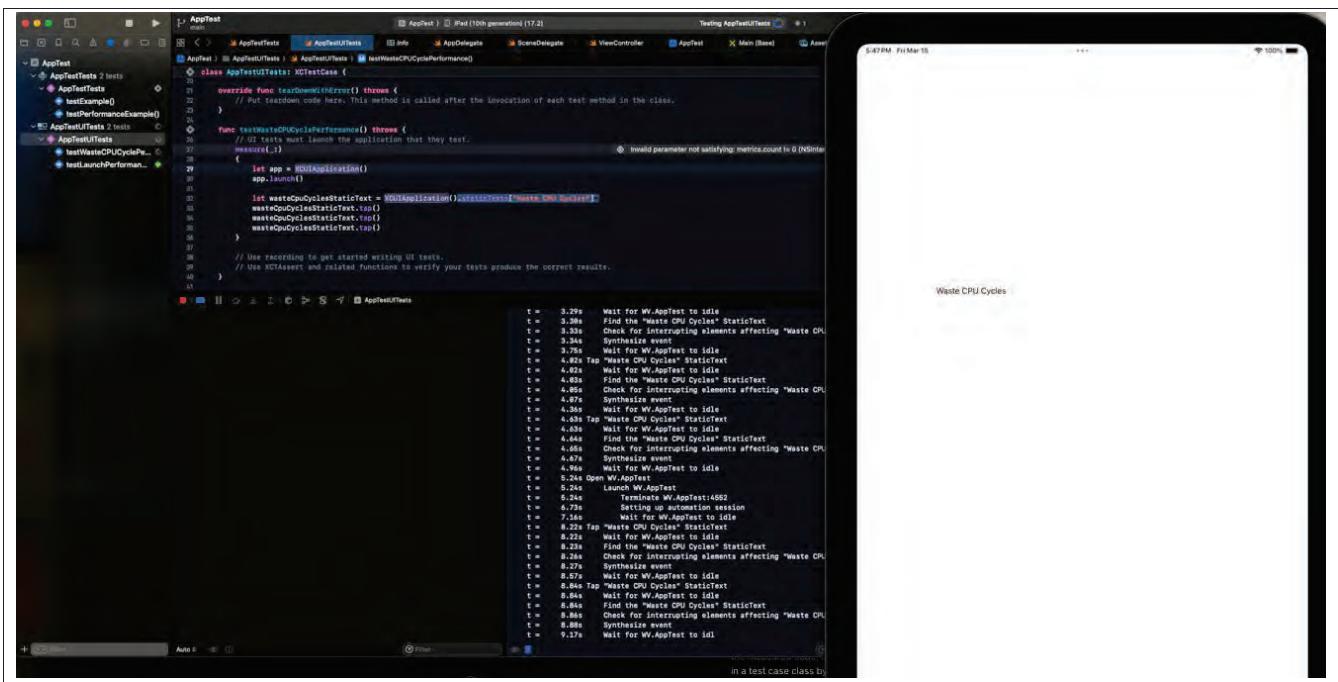
[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.



Test scripts can then be written (in code) or recorded for later execution. For example, screenshots of a performance test which repeatedly presses a button in a sample application are shown below:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

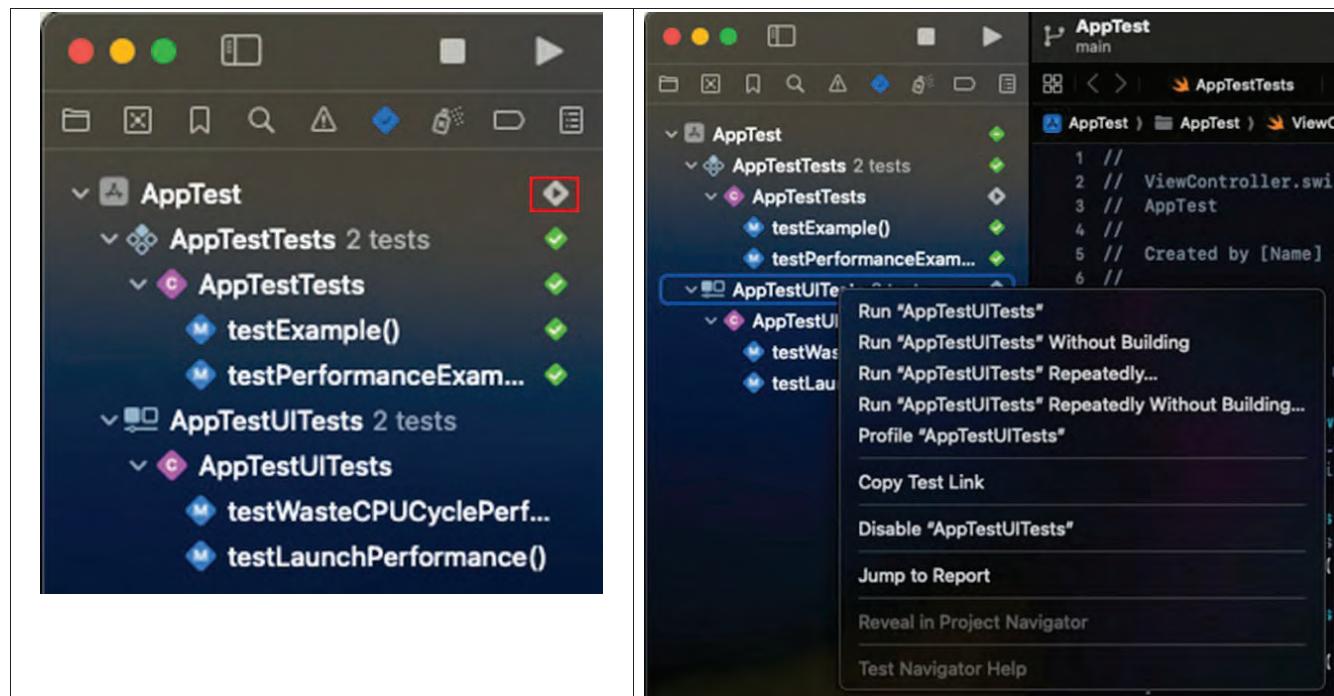
[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.



The tests can be executed by clicking the run icon next to the tests or by opening the test menu and selecting to run the tests:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

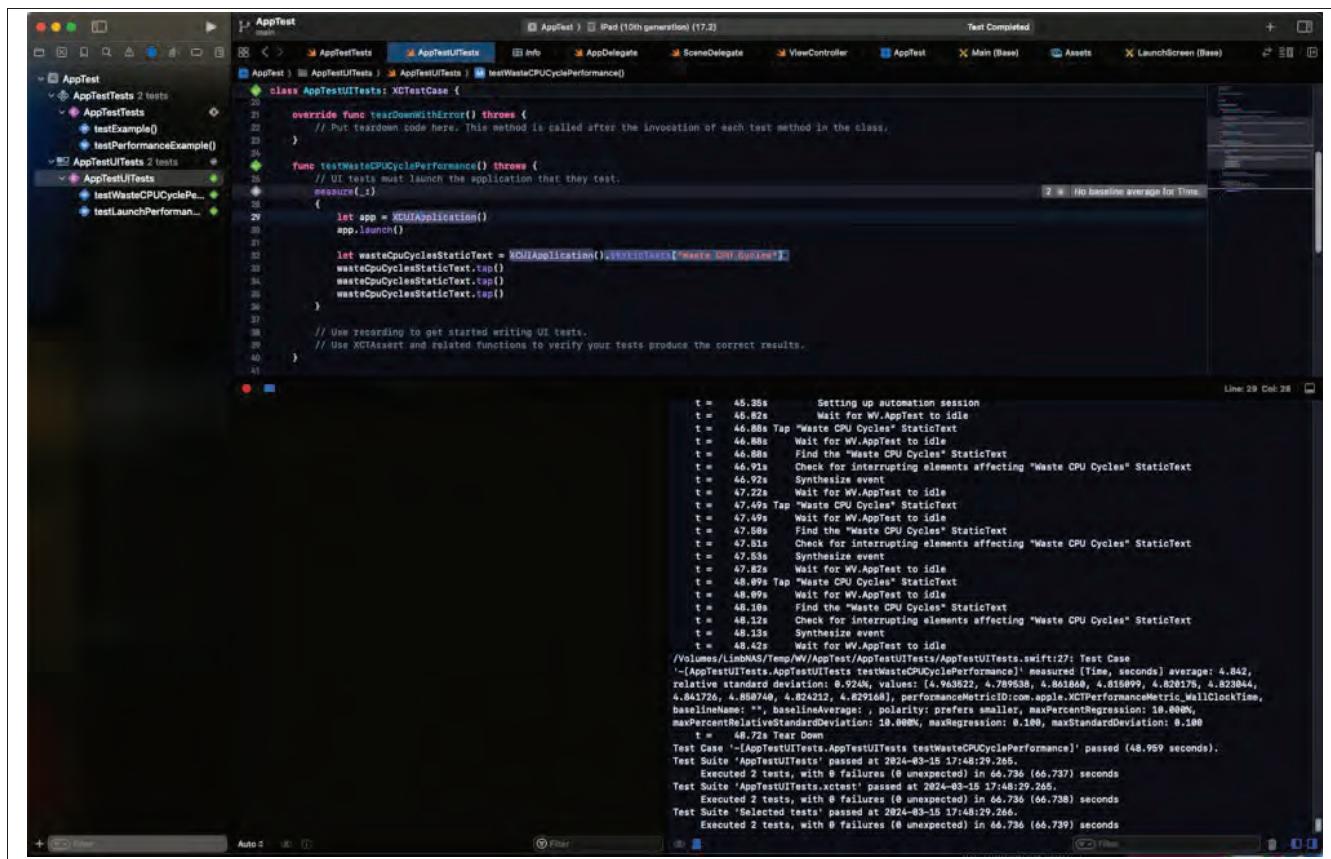
[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.



The results of the testing are shown after the test completes:

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.



The screenshot shows the Xcode interface during a test run. The left sidebar displays a tree view of test cases under 'AppTest'. The main editor window shows the Swift code for the 'testWasteCPUcyclePerformance' test. The bottom right pane shows the test results, including a timeline of events and a summary of the test case passing.

```

class AppTestUITests: XCTestCase {
    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testWasteCPUcyclePerformance() throws {
        // UI tests must launch the application that they test.
        measure {}

        let app = XCUIApplication()
        app.launch()

        let wasteCpuCyclesStaticText = XCUIApplication().staticTexts["Waste CPU Cycles"]
        wasteCpuCyclesStaticText.tap()
        wasteCpuCyclesStaticText.tap()
        wasteCpuCyclesStaticText.tap()
    }

    // Use recording to get started writing UI tests.
    // Use XCTAssert and related functions to verify your tests produce the correct results.
}

```

Test Suite 'AppTestUITests' passed at 2024-03-15 17:48:29.265.
 Executed 2 tests, with 0 failures (0 unexpected) in 66.736 (66.737) seconds
 Test Suite 'AppTestUITests.xctest' passed at 2024-03-15 17:48:29.265.
 Executed 2 tests, with 0 failures (0 unexpected) in 66.736 (66.738) seconds
 Test Suite 'Selected tests' passed at 2024-03-15 17:48:29.266.
 Executed 2 tests, with 0 failures (0 unexpected) in 66.736 (66.739) seconds

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[8] The system of claim 1, wherein the software is further configured to create one or more scenarios that include scripts that impact either the performance of the application, or the network, or both.

The scenarios that include scripts further impact either the performance of the application, or the network, or both. As shown in the screenshots above, execution of the test scripts (such as “testWasteCPUCyclePerformance”) impacts the performance of the application by triggering certain code the script is intended to test, and Xcode shows the results of that impact in regard to the amount of time it took to execute. Scripts that result in the application creating network communications (such as, for example, requesting data from a server using a protocol such as TCP/IP or UDP) also impact the performance of the network (or both the application and the network).

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.

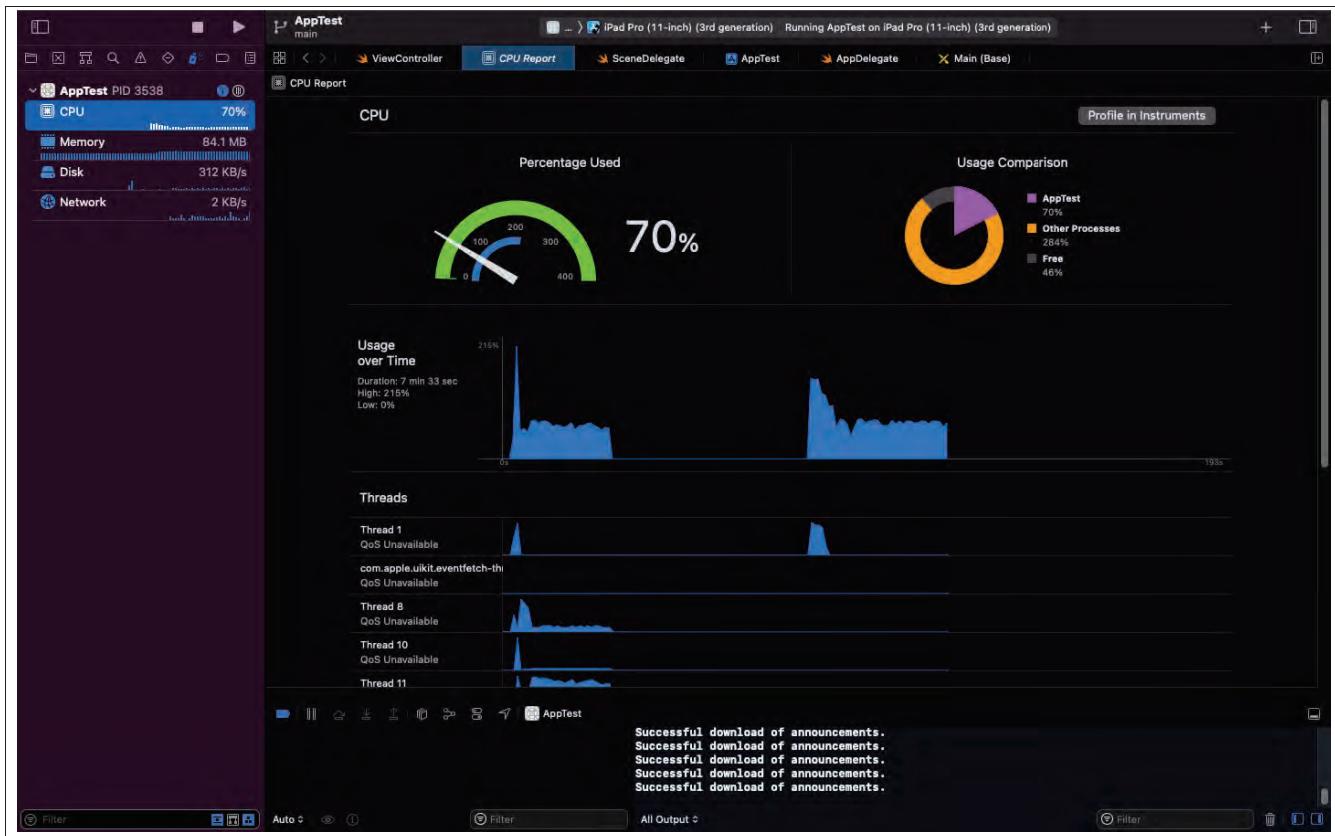
Claim 13

[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.

As discussed above for limitation [1a].2 (and incorporated here by reference), the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both. For example, the graphical displays in both Xcode IDE and Instruments enable a user to identify either application performance, or network performance, or both. Examples of this are shown in, for example, the non-limiting CPU and Networking graph examples below:

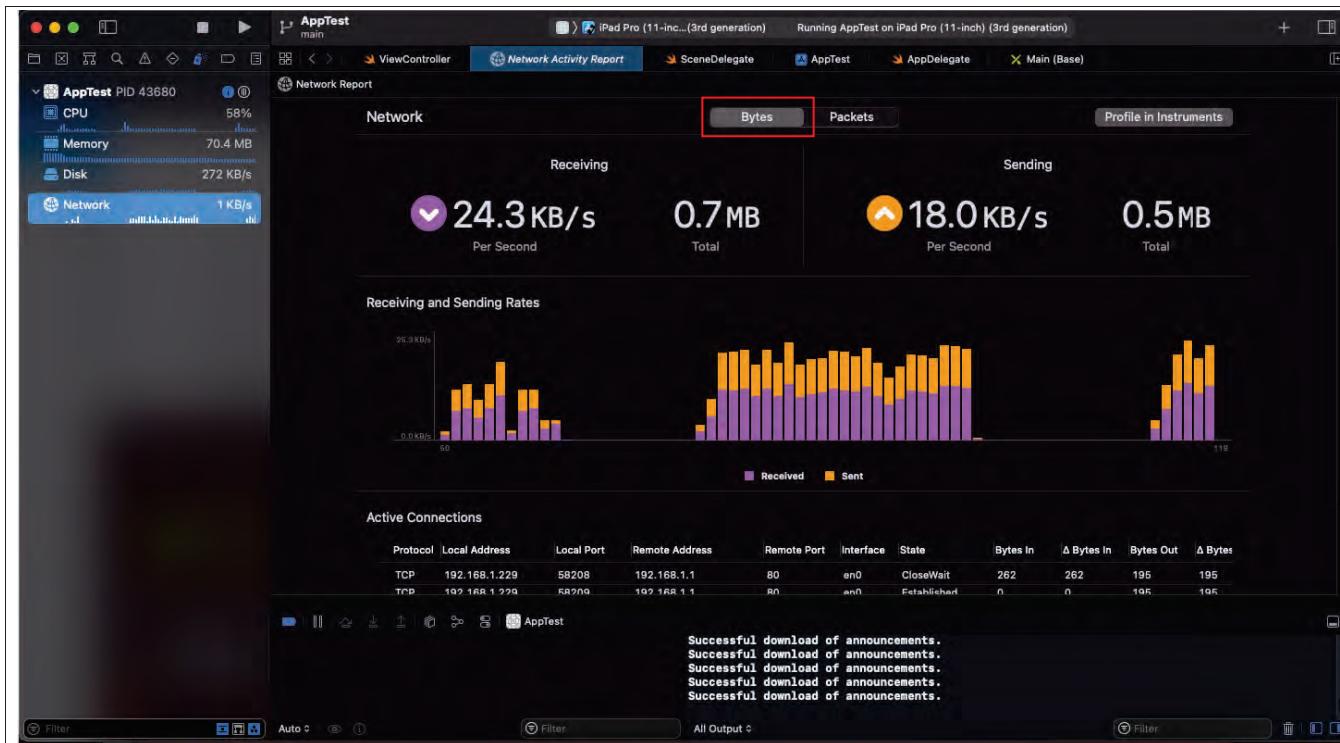
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.



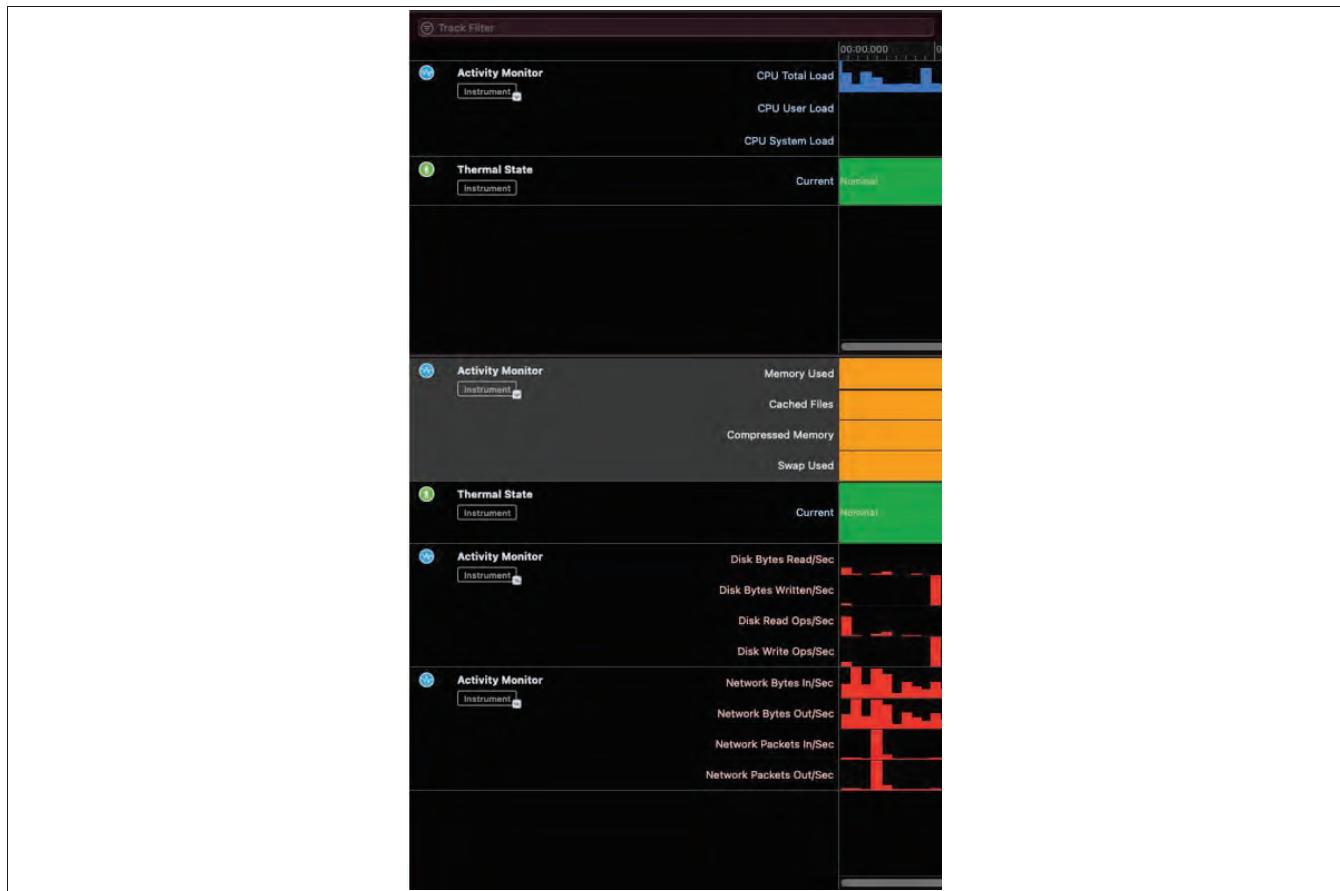
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[13] The system of claim 1, wherein the software is further configured to display data graphically which is configured to enable a user to identify either application performance, or network performance, or both.

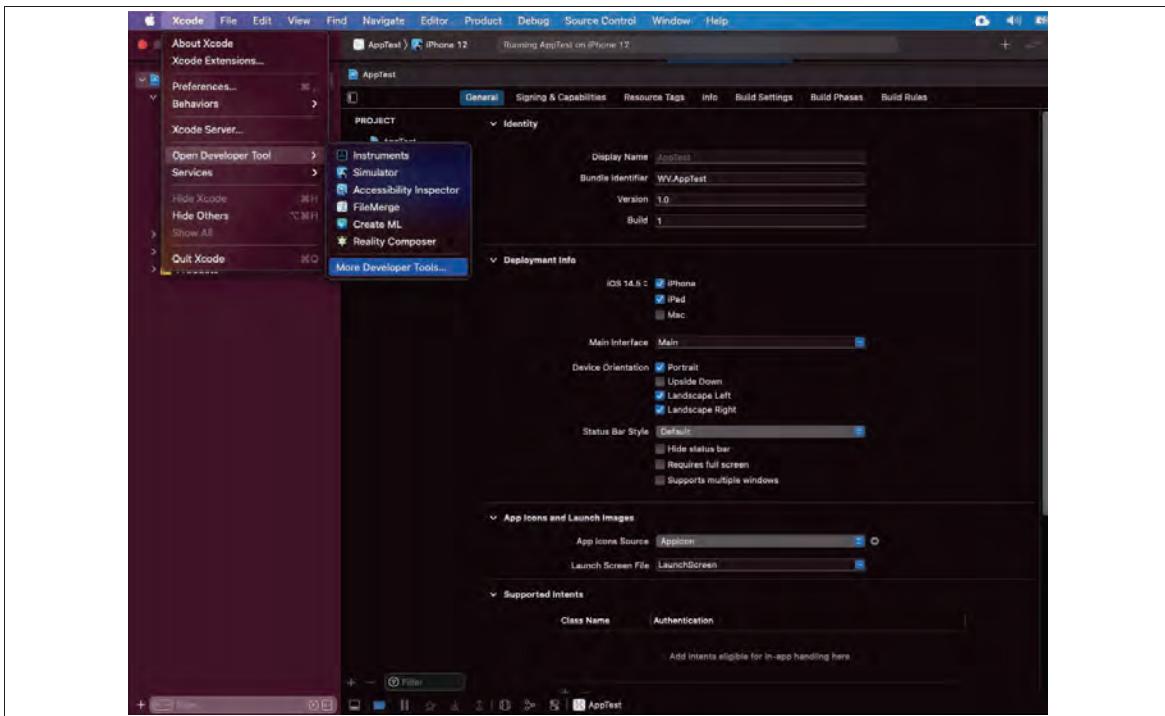


Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[14] The system of claim 1, wherein the software is further configured to interact with a network to enable a user to update the software

Claim 14

[14] The system of claim 1, wherein the software is further configured to interact with a network to enable a user to update the software



The above image shows the Xcode IDE. A user can interact with a network to enable a user to update the software by selecting Xcode->Open Developer Tool->More Developer Tools...

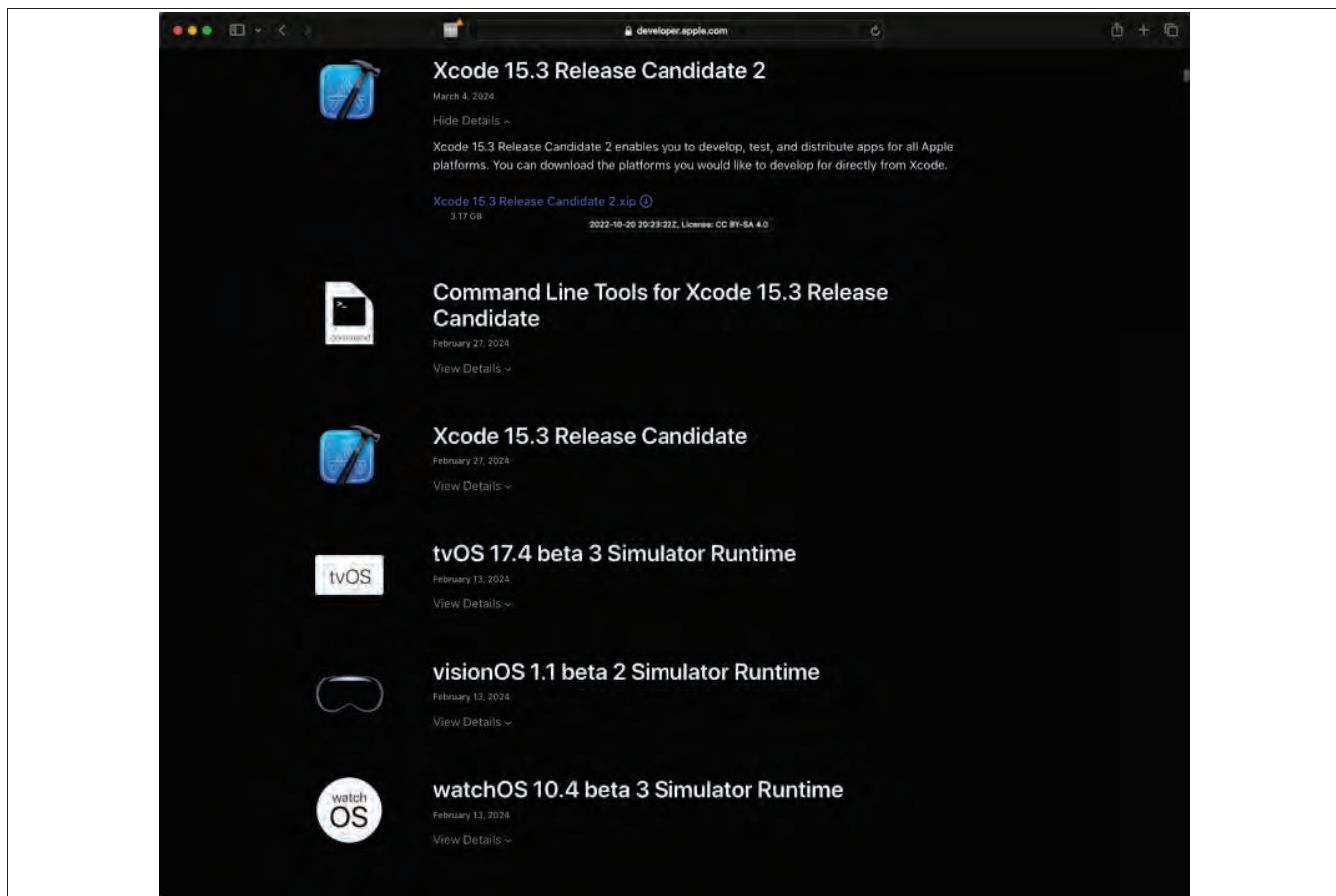
Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[14] The system of claim 1, wherein the software is further configured to interact with a network to enable a user to update the software

Selecting “More Developer Tools...” opens a web browser to a web page which enables a user to update the software to a newer version (in this example, Xcode 15.3 Release Candidate 2):

Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[14] The system of claim 1, wherein the software is further configured to interact with a network to enable a user to update the software



Infringement Chart for U.S. Patent No. 9,298,864 (Apple's Xcode Developer Tools)

[19] The system of claim 18, wherein the network profiles are configured to enable a user to either manage the profiles, or create custom profiles, or both.

Claim 17

[17] The system of claim 1, wherein the software is further configured to enable a user to select from one or more network characteristics for testing how well mobile content performs on the mobile device.

As discussed above for limitation [1a].1 (and incorporated here by reference), the software is further configured to enable a user to select from one or more network characteristics (including, *inter alia*, in/out bandwidth, in/out packet loss, in/out delay, and DNS delay) for testing how well mobile content (such as that generated by the application being tested) performs on the mobile device.